

UNIVERSITETET I OSLO
Institutt for informatikk

**Prognose ved
"Local Binary
Pattern"
teksturanalyse av
cellekjerner fra
prostata- og
eggstokkreft**

Caroline Skudal &
Petter Aarseth

2. mai 2013



Prognose ved "Local Binary Pattern"
teksturanalyse av cellekjerne fra prostata- og
eggstokkreft

Caroline Skudal & Petter Aarseth

2. mai 2013

Forord

Denne oppgaven er utført ved det matematisk-naturvitenskaplig fakultet ved Universitetet i Oslo som en del av graden Master of Science i Informatikk. Studiene ble startet i august 2011 og ferdigstilt i mai 2013. Arbeidet har blitt utført ved Institutt for Informatikk ved Universitetet i Oslo og Institutt for Medisinsk Informatikk ved Radiumhospitalet. Arbeidet er utført i samarbeid mellom Caroline Skudal og Petter Aarseth.

Takk

Vi ønsker å takke vår hovedveileder Professor Fritz Albrechtsen for alle bidrag, forslag til forbedring og hjelp vi har fått igjennom hele masterperioden. Vi ønsker også å takke våre med-veiledere Professor Håvard E. Danielsen og Dr. Birgitte Nielsen for deres hjelp, støtte og veiledning. Vi takker også Tarjei Sveinsgjerd Hveem for all hjelp med detaljer rundt M98 og testing på et utvidet utvalg av M98.

I tillegg ønsker vi å takke Andreas Kleppe for å komme med idéer underveis, samt Rolf Anders Syvertsen for å sette oss inn i bildematerialet, og hvordan man i praksis henter ut slike bilder, og i tillegg takk til alle ved Institutt for Medisinsk Informatikk for deres hyggelige mottagelse av oss. Vi ønsker også å takke våre familier for deres støtte og tålmodighet.

Sammendrag

Ved behandling av kreft er det ofte forskjell på hvordan pasienter med samme krefttype reagerer på lik behandling, selv i tilfeller hvor pasientene har lik alder og medisinsk historie. Vi ønsker å forutse hvordan en pasient diagnostisert med kreft reagerer på en type behandling, og om vedkommende vil få tilbakefall etter behandlingen. Vanskeligheten er at flertallet av cellekjernene til en pasient med dårlig prognose oppfører seg som cellekjernene til en pasient med god prognose [15].

I denne oppgaven har vi anvendt lokal teksturanalyse på prostatakreft og eggstokkreft for å skille mellom pasienter med god og dårlig prognose. Vi har benyttet Lokalt Binært Mønster, "Local Binary Pattern" (LBP), lokal varians og kombinasjoner av de to nevnte operatorene på bilder av cellekjerne fra prostata og eggstokk. I tillegg har vi satt opp en bildepyramide for å se om det å endre oppløsning på cellekjerne-bildene kan bedre resultatene. Vi har også benyttet adaptive egenskaper basert på forekomstmatriser satt sammen av resulterende verdier fra LBP og andre operatorene. Operatorene vi har benyttet sammen med LBP er entropi, varians og senterpikselens gråtone.

Det beste resultatet vi oppnådde var på opp mot 74 % korrekt klassifiseringsrate i test. Dette er 4 prosentpoeng bedre enn det Nielsen og Danielsen oppnådde i [50], på tilsvarende eggstokk-materiale som er brukt her. De beste resultatene ble oppnådd ved å bruke LBP med 8 piksels nabolag på nedskalerte cellekjerne-bilder i eggstokk-materialet. Generelt observerte vi at begge operatorene (LBP og Varians) med 8 piksels nabolag, benyttet på nedskalerte cellekjerne-bilder ga de beste resultatene.

Abstract

In the treatment of cancer, patients with the same type of cancer often react differently, even when the patients have the same age and medical history. Our wish is to predict how a patient diagnosed with cancer will react on a given treatment, and whether the patient will relapse after the treatment. The difficulty is that the majority of the cell nuclei of a patient with bad prognosis will behave like the cell nuclei of a patient with good prognosis [15].

In this thesis, we have applied local texture analysis on prostate and ovarian cancer to separate patients with good and bad prognosis. We have used Local Binary Pattern (LBP), local variance and combinations of the two mentioned operators on images of cell nuclei from prostate and ovary. In addition, we have created an image pyramid, to see whether reduction of the resolution of the cell nuclei images can improve the results. We have also used adaptive features based on occurrence matrices from LBP and three other operators. The other operators were entropy, local variance and gray level.

At best, we achieved a correct classification rate of about 74 %. This result is 4 percentage points better than what Nielsen and Danielsen achieved in [50] on the same ovarian material. The best results were achieved by using LBP on images with reduced image resolution. In general, the best results were obtained by using LBP and local variance on images with reduced image resolution.

Innhold

1	Introduksjon	1
1.1	Grunnlag	2
1.2	Påstand	2
1.3	Agenda	3
1.4	Bruk av referanser	3
1.5	Strukturering av oppgaven	4
2	Metoder - Verktøy	5
2.1	Teksturanalysemetoder med vekt på Lokalt Binært Mønster .	5
2.1.1	Gråtone Samforekomst-Matrise (GLCM)	6
2.1.2	Gråtonenivå Sum- og Differansehistogram	7
2.1.3	Gråtone Løpelengde-Matrise (GLRLM)	8
2.1.4	Samforekomst av Gråtone Løpelengde-Matrise (CGL- RLM)	8
2.1.5	Gråtone Entropi-Matrise (GLEM)	9
2.1.6	Laws' tekstur-energimålere	9
2.1.7	Lokalt Binært Mønster (LBP)	10
2.2	Distansemålere	17
2.2.1	Mahalanobis distanse	17
2.2.2	Bhattacharyya	19
2.2.3	Chi-kvadratfordeling	20
2.2.4	"Earth Mover's Distance"	20
2.2.5	Andre distansemålere	21
2.2.6	Oppsummering	21
2.3	Nedskalering og bildepyramide	22
2.3.1	Nedskalering	23
2.3.2	Bildepyramider	25
2.4	Klassifikatorer	29
2.4.1	Minimum distanse klassifisering	29
2.4.2	Nærmeste nabo klassifisering	29
2.4.3	Bayesisk Klassifisering	31
2.4.4	Støttevektormaskin (SVM)	31
2.4.5	Oppsummering	32
3	Prosjektet - datasettet	33
3.1	Datasettene	33
3.1.1	Prostata-materialet	33

3.1.2	Eggstokk-materialet	35
3.2	Tidligere resultater	36
4	Design av eksperimentet	39
4.1	LBP benyttet på nabolag bestående av 8 piksler i originale cellekjerne-bilder	39
4.1.1	Fordelingen av de ulike TU-numrene	41
4.1.2	Reduksjon av antall TU-numre	46
4.1.3	Sannsynlighetsfordeling for TU-gruppene	46
4.1.4	Varians	51
4.1.5	Kvantisering av varians	51
4.1.6	Valg av distansemåler for TU	55
4.1.7	Klassifikator	55
4.1.8	Kort om visualisering av klassene	56
4.1.9	Valg av distansemåler for varians	56
4.2	LBP benyttet på nabolag bestående av 16 piksler	58
4.3	LBP benyttet på nabolag bestående av 8 piksler i nedskalerte cellekjerne-bilder	58
4.3.1	Interpolasjon	59
4.3.2	Skalering	59
4.4	Adaptive egenskaper	60
4.4.1	Kombinasjon mellom senterpikselens gråtone og nabo- lagets TU-gruppe	60
4.4.2	Kombinasjon mellom variansen i det lokale nabolaget og nabolagets TU-gruppe	62
4.4.3	Kombinasjon mellom entropien i det lokale nabolaget og nabolagets TU-gruppe	63
4.4.4	Valg av kombinasjon	65
4.4.5	Bruk av matrisene - Adaptive egenskaper	69
5	Resultater og Diskusjon	75
5.1	Resultater fra operatører benyttet på nabolag bestående av 8 piksler i originale cellekjerne-bilder	76
5.1.1	Resultater fra bruk av LBP	76
5.1.2	Resultater fra bruk av varians	78
5.1.3	Kombinering av egenskaper	78
5.2	Resultater fra operatører benyttet på nabolag bestående av 16 piksler i originale cellekjerne-bilder	81
5.2.1	Resultater fra bruk av LBP	82
5.2.2	Resultater fra bruk av varians	82
5.2.3	Kombinering av egenskaper for klassifisering	83
5.3	Resultater fra operatører benyttet på nabolag bestående av 8 piksler i nedskalerte cellekjerne-bilder	86
5.3.1	Resultater fra bruk av LBP	86
5.3.2	Resultater fra bruk av varians	87
5.3.3	Kombinering av egenskaper	88
5.4	Diskusjon rundt datasettet M98	90
5.4.1	Operatører på 8 piksels nabolag i originalbildene	90

5.4.2	Operatorer benyttet på 16 piksels nabolag i originalbildene	93
5.4.3	Operatorer benyttet på 8 piksels nabolag i de nedskalerte bildene	93
5.4.4	Kombinering av egenskaper	95
5.4.5	Litt mer generell diskusjon rundt M98	98
5.5	Resultater fra bruk av operatorer benyttet på nabolag bestående av 8 piksler på originale bilder fra L23	101
5.5.1	Resultater fra bruk av LBP	101
5.5.2	Resultater fra bruk av varians	101
5.5.3	Kombinering av egenskaper	102
5.6	Resultater fra bruk av operatorer benyttet på nabolag bestående av 16 piksler på originale bilder fra L23	103
5.6.1	Resultater fra bruk av LBP	104
5.6.2	Resultater fra bruk av varians	104
5.6.3	Resultater etter kombinering	105
5.7	Resultater fra bruk av operatorer benyttet på nabolag bestående av 8 piksler på nedskalerte bilder fra L23	107
5.7.1	Resultater fra bruk av LBP	107
5.7.2	Resultater fra bruk av varians	108
5.7.3	Resultater etter kombinering	109
5.8	Diskusjon rundt datasettet L23	110
5.8.1	Operatorer benyttet på 8 piksles nabolag i originalbildene	110
5.8.2	Operatorer benyttet på 16 piksels nabolag i originale cellekjerne-bilder	115
5.8.3	Operatorer benyttet på 8 piksles nabolag i nedskalerte cellekjerne-bilder	117
5.8.4	Oppsummering av hva vi har sett ved analyse av L23	120
5.9	Resultater fra bruk av adaptive egenskaper benyttet på nabolag bestående av 8 piksler på originale bilder fra M98	122
5.9.1	Kombinasjon av TU og entropi	122
5.9.2	Kombinasjon av TU og varians	122
5.9.3	Kombinasjon av TU og senterpikselens gråtone	122
5.10	Diskusjon rundt adaptive egenskaper benyttet på M98	124
5.10.1	Kombinasjon mellom TU og entropi	124
5.10.2	Kombinasjon mellom TU og varians	125
5.11	Resultater fra bruk av adaptive egenskaper benyttet på nabolag bestående av 8 piksler på originale bilder fra L23	126
5.11.1	Kombinasjon av TU og entropi	126
5.11.2	Kombinasjon av TU og varians	126
5.11.3	Kombinasjon av TU og gråtone	127
5.12	Diskusjon rundt adaptive egenskaper benyttet på de originale bildene i L23	128
5.12.1	Kombinasjon mellom TU og entropi	128
5.12.2	Kombinasjon mellom TU og varians	133
5.12.3	Kombinasjon mellom TU og senterpikselens gråtone	133

5.13	Resultater fra bruk av adaptive egenskaper benyttet på nabolag bestående av 8 piksler på nedskalerte bilder fra L23 .	136
5.13.1	Kombinasjon av TU og entropi	136
5.13.2	Kombinasjon av TU og varians	136
5.13.3	Kombinasjon av TU og senterpikselens gråtone	137
5.14	Diskusjon rundt adaptive egenskaper benyttet på nedskalerte bilder i L23	138
5.14.1	Kombinasjon mellom TU og entropi	138
5.14.2	Kombinasjon mellom TU og varians	141
5.15	Oppsummering rundt bruken av adaptive egenskaper	146
6	Konklusjon og videre arbeid	149
7	Vedlegg	153
7.1	Vedlegg A	153
7.2	Vedlegg B	158
7.3	Vedlegg C	171

Kapittel 1

Introduksjon

I denne oppgaven har vi analysert cellekjerner i kreftsvulster fra både prostata og eggstokk. Det er ofte forskjell på hvordan pasienter med samme type kreft reagerer på lik behandling, selv i tilfeller hvor disse pasientene har lik alder og medisinsk historie. Av den grunn ønsker man å finne ut hvordan en pasient som er diagnostisert med kreft kommer til å reagere på en type behandling, og eventuelt om vedkommende vil få tilbakefall etter behandlingen. Vi deler derfor opp pasientene i to grupper. Pasientene som reagerer positivt på behandlingen, og som ikke får tilbakefall sier vi har god prognose, mens de resterende pasientene blir sagt å ha dårlig prognose. Pasientene vi har sett på i denne oppgaven, og som har overlevd, har blitt fulgt opp i minst ti år etter behandling, og vi antar at eventuelle tilbakefall etter behandling ville ha forekommet i løpet av denne tidsperioden.

Ettersom man tenker seg at alle typer kreft stammer fra genomiske endringer, ser man for seg at det er mulig å se ulikheter mellom god og dårlig prognose ved å se på grupperinger av gener [46]. Slike grupperinger vil igjen være korrelert til endringer i cellekjernes kromatin. Dette gir utslag i cellekjernes tekstur, noe vi kan måle ved hjelp av teksturanalyse. Selv om vi i denne oppgaven begrenser oss til statistiske teksturanalytiske metoder, finnes det fortsatt veldig mange ulike metoder å velge mellom. Det er tydelig fra mange ulike studier, som Nielsen et al. gjennomgår i [46], at statistisk teksturanalyse helt klart har noe for seg.

I denne oppgaven har vi sett på noen utvalgte statistiske metoder, som har blitt benyttet på monolayer-bilder av cellekjerner til pasienter. Her er hovedvekten lagt på en gråtone- og rotasjonsinvariant operator kalt Lokalt Binært Mønster, "Local Binary Pattern" (LBP), men vi har også sett på andre metoder som kan bidra til å øke klassifiseringsraten ved kombinasjon med LBP. Grunnen til at vi har valgt å legge vekten på LBP er at LBP har vist seg å være en god operator i statistisk teksturanalyse [54] [56] [36] [2] [18], men den har ikke tidligere blitt brukt på cellekerne-bilder som et prognostisk verktøy. Siden man har sett at statistiske teksturanalysemetoder har noe for seg ved analyse av kreftcellekjerner [46], ser vi ingen grunn til at LBP ikke skal gjøre det bra på dette området, når operatoren gjør det

bra på flere andre områder.

En LBP-operator benytter et forhåndsbestemt nabolag, som består av piksler jevnt fordelt i en sirkel rundt en senterpiksel, og er i mange tilfeller de 8 nærmeste pikselene til senterpikselen [56]. Et slikt 8 piksels nabolag fører til at vi ser på en veldig liten andel av cellekjernen, og dette området vil ikke nødvendigvis være tilstrekkelig stort til å beskrive den lokale teksturen. Av den grunn har vi benyttet to metoder for å se på en større andel av cellekjernen om gangen. Den første metoden går ut på å la operatoren bli basert på et større nabolag, mens den andre metoden går ut på å bruke bildepyramider.

Ved egenskapsutvelgelse benytter vi to ulike fremgangsmåter. Den ene baseres på å finne klassiske egenskaper (predefinerte egenskaper) som tilsynelatende vil gjøre det bra, mens den andre går ut på å benytte såkalte adaptive egenskaper. Tanken bak adaptive egenskaper er at man vektlegger de bildepunkter eller egenskapene som ser ut til å inneha mest klasseskillende informasjon.

1.1 Grunnlag

”In digital pathology, the field of nuclear texture analysis gives information about the spatial arrangement of the pixel gray levels in a digitized microscopic nuclear image, providing statistical texture measures that may be used as quantitative tools for diagnosis and prognosis of human cancer”. Slik starter Nielsen et al. sammendraget i [46], og dette gir et godt bilde av grunnlaget til denne masteroppgaven.

I denne oppgaven benytter vi en teksturoperator kalt Lokalt Binært Mønster, ”Local Binary Pattern” (LBP) i et forsøk på å skille pasienter med god og dårlig prognose av kreft i prostata og eggstokk. He og Wang [28] skriver at forgjengeren til LBP ”extract textural information of an image with a more complete respect of texture characteristics”. I [28] sammenlikner He og Wang operatoren med blant annet statistikk basert på GLCM, som kun ser på pikselpar, mens deres egen operator ser på 8 piksler. Dette gjelder også for LBP. Ojala et al. [61] tok skrittet videre fra He og Wangs operator, og kalte den ”Local Binary Pattern”. I [54] trekker Ojala et al. frem fordelene med at operatoren er ”very robust in terms of gray-scale variations since the operator is, by definition, invariant against any monotonic transformation of the gray-scale”.

1.2 Påstand

I denne oppgaven har vi benyttet teksturanalyse i form av LBP på bilder av cellekjerner i prostata- og eggstokkreft. Vi hevder at metoden virker lovende til å skille mellom pasienter med god og dårlig prognose, og er minst like suksessfull som andre og høyere ordens statistiske metoder. I tillegg ender

vi opp med en fremgangsmåte som er langt enklere enn flere metoder som benyttes i dag.

1.3 Agenda

I denne oppgaven har vi sett på åtte ulike fremgangsmåter. Tre av dem går ut på å beregne LBP, og eventuelt andre operatorer kombinert med førstnevnte, på cellekjerne-bilder fra prostata- og eggstokkreft. Forskjellen mellom de tre fremgangsmåtene ligger i nabolagsstørrelsen til operatorene, og i tillegg om man benytter operatorene på de originale cellekjerne-bildene, eller nedskalerte cellekjerne-bilder. Etter at operatorene har blitt benyttet har vi satt opp sannsynlighetsfordelinger over utregnede verdier fra operatorene, både for hver enkelt pasient og for prognosegruppene. Her er en prognosegruppe gruppen med pasienter som har lik prognose. Deretter har avstanden mellom en pasients sannsynlighetsfordeling og sannsynlighetsfordelingene til prognosegruppene blitt beregnet.

De fem andre fremgangsmåtene går ut på å beregne adaptive egenskaper, for deretter å benytte disse i klassifiseringen. To av disse fremgangsmåtene baserer de adaptive egenskapene på forekomster av ut-verdier fra LBP kombinert med entropi, to andre fremgangsmåter baserer de adaptive egenskapene på forekomster av ut-verdier fra LBP kombinert med varians, mens den siste baserer de adaptive egenskapene på ut-verdier fra LBP og senterpikselens gråtone. Kombinasjonene med entropi og varians blir benyttet på både originale og nedskalerte cellekjerne-bilder, mens kombinasjonen med senterpikselsens gråtone blir kun benyttet på originalbildene. Disse forekomstene blir så brukt for å beregne de adaptive egenskapene. Deretter blir sannsynlighetsfordelinger for hver pasient og prognosegruppe satt opp basert på de adaptive egenskapene, før vi finner avstander mellom pasientenes og de to prognosegruppens sannsynlighetsfordelinger, slik vi gjorde ved bruk av de ikke-adaptive egenskapene.

1.4 Bruk av referanser

Vi har hovedsakelig brukt publisert artikler og utvalgte lærebøker som referanser. Vi har noen referanser fra Wikipedia, som viser definisjoner og sammenhenger mellom aktuelle begreper, samt anvendelser. Av den grunn anser vi disse artiklene som mer informative enn mulige lærebøker med korte definisjoner. I tillegg har veiledere bekreftet at innholdet i disse artiklene er korrekt, og godkjent dem til kildebruk. Ved bruk av bilder, som vi ikke har laget selv, refererer vi til et par bildedatabaser, samt lærebøker/artikler og enkelte foiler fra forelesninger i INF4300, som ble holdt høsten 2012 ved Instituttet for Informatikk ved Universitetet i Oslo. Til slutt har vi også benyttet våre veiledere som referanse på noen områder, og der det er aktuelt har vi oppgitt dette.

1.5 Strukturering av oppgaven

Kapittel 2 omhandler metoder og verktøy, og i dette kapitlet vil man først få en gjennomgang av teksturanalysemetoder med vekt på LBP. Deretter får man en gjennomgang av ulike distansemålere. Neste seksjon omhandler nedskalering og bildepyramider. Her vil vi ta for oss ulike nedskaleringsmetoder og forskjellige bildepyramider. Kapitlet blir avsluttet med en gjennomgang av ulike klassifikatorer. Det neste kapitlet (kapittel 3) omhandler de to datasettene vi benytter. Kapittel 4 tar for seg designet av prosjektet, og starter med å se på de ulike fremgangsmåtene for bruk av LBP og lokal varians. Deretter avsluttes det med fremgangsmåten for bruk av adaptive egenskaper basert på kombinasjoner mellom LBP og andre operasjoner.

I kapittel 5, som omhandler resultater og diskusjon, har vi valgt å ta en gjennomgang av resultatene for prostata-materialet med etterfølgende diskusjon. Deretter kommer en gjennomgang av resultatene for eggstokk-materialet med etterfølgende diskusjon. Etter dette kommer resultater fra bruk av adaptive egenskaper på prostata-materialet, som blir etterfulgt av en diskusjon omkring resultatene. Resultatene fra de adaptive egenskapene brukt på de originale bildene fra eggstokk-materialet, og tilhørende diskusjon, kommer etter prostata-materialet. De siste resultatene kommer fra de adaptive egenskapene brukt på de nedskalerte bildene fra eggstokk-materialet, og blir også etterfulgt av en egen diskusjon. Selv om vi er klar over at det er vanlig å ha et kapittel til alle resultater og et annet til diskusjon har vi, på grunn av den omfattende mengden resultater, valgt å organisere det på denne måten for å øke lesbarheten. Kapitlet blir avsluttet med en mer generell diskusjon. Oppgaven avsluttes med kapittel 6, som inneholder konklusjon og forslag til videre arbeid.

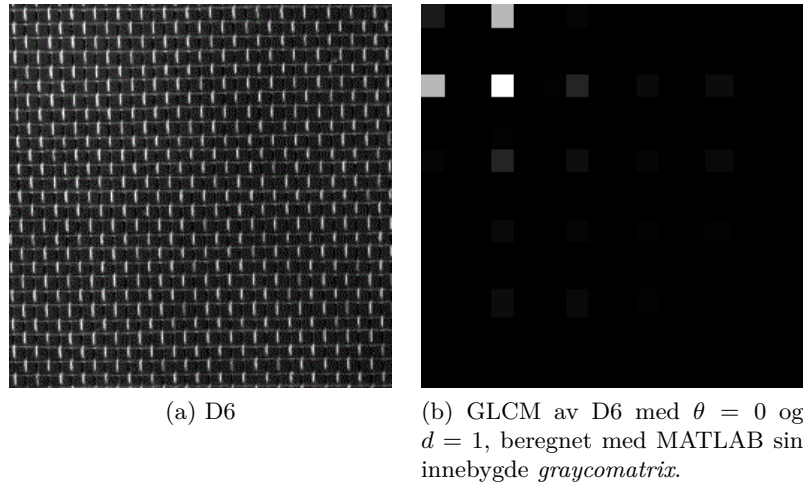
Kapittel 2

Metoder - Verktøy

Ved beregning av kreftprognose basert på teksturanalyse av cellekjerner i prostata og eggstokk, er det mange ulike verktøy man kan benytte seg av. I dette kapitlet går vi gjennom slike verktøy for teksturanalysen, målingen av ulikhetene mellom klassene, bruk av bilder samt klassifiseringen.

2.1 Teksturanalysemetoder med vekt på Lokalt Binært Mønster

I digital bildeanalyse er det ofte hensiktsmessig å kunne skille forskjellige områder i bilder fra hverandre basert på lokale egenskaper i disse områdene. Innenfor medisin er det for eksempel nyttig å kunne tydeliggjøre hvorvidt celler har forskjellig tekstur, og på den måten gjøre det lettere å stille diagnoser eller skille mellom forskjellige prognoser. I [46] tar Nielsen et al. for seg statistiske metoder for å beskrive tekstur. Slike metoder beskriver egenskaper ved teksturen, ved å se på piksler i lokale områder, og kan variere fra første ordens statistikk, som avhenger av en piksel, til andre- og høyere ordens statistikk, som avhenger av to eller flere piksler. Første ordens statistikk baserer seg på gråtonehistogrammet til et lokalt område, og kan for eksempel være forventningsverdien eller variansen i området. Andreordens statistikk beskriver relasjonen mellom pikselpar. "Gray-Level Cooccurrence Matrices" (GLCM) er et eksempel på dette. Høyere ordens statistikk beskriver relasjoner mellom tre eller flere piksler og inkluderer blant annet "Gray-Level Run Length Matrices" (GLRLM) og Laws' tekstur-energimålere. I tillegg til statistiske metoder har man også de som kalles strukturelle metoder [25]. Når man tar for seg strukturelle metoder, kan man tenke seg at en tekstur er satt sammen av forskjellige tekstler på en karakteristisk måte for denne teksturen. Ved å beskrive denne sammensetningen, kan man også beskrive selve teksturen. Her kommer vi imidlertid til å fokusere på andre- og høyere-ordens statistiske metoder, deriblant den mye anvendte metoden GLCM. I tillegg blir også Laws' tekstur-energimålere og generaliseringer av disse sett på, før det blir rundet av med en mer nøyaktig gjennomgang av Lokalt Binært Mønster, "Local Binary Pattern" (LBP).



Figur 2.1: Eksempel på GLCM til et bilde fra Brodatz teksturer [7]. Figurene er hentet fra [46].

2.1.1 Gråtone Samforekomst-Matrise (GLCM)

En av de mest brukte metodene for å beregne 2. ordens statistiske teksturegenskaper er "Gray-Level Cooccurrence Matrix", GLCM. Dette er i følge Nielsen et al. [46] en av de mest slagkraftige metodene for generell teksturklassifikasjon. Som Haralick et al. [26] påpeker er GLCM en symmetrisk matrise hvor antall rader og kolonner er lik antall gråtoner G , i bildet. Matriseelementet $P(i,j|d,\theta)$ er sannsynligheten for gråtoneovergang mellom gråtone i og j , med gitt distanse d og vinkel θ . Ettersom antall gråtoner bestemmer størrelsen på GLCM, kan det ofte være en fordel å redusere antall gråtoner, og på den måten unngå for mange tomme plasser i matrisen. En slik reduksjon bidrar også til å gjøre GLCM mer stabil, og dermed også mindre sensitiv til størrelsen på teksturutvalget. GLCM regnes ut ved å gå gjennom bildet, med en gitt distanse d og retning θ , og underveis telle opp forekomsten av hver enkelt gråtoneovergang. Det er vanlig å beregne GLCM for $\theta = 0, 45, 90$ og 135 grader. Symmetrisk GLCM oppnås ved å legge sammen matrisen med dens transponerte. For å gjøre GLCM rotasjonsinvariant kan man beregne en isotropisk versjon. Den isotropiske versjonen oppnår man ved å ta gjennomsnittet av GLCM med $\theta = 0, 45, 90$ og 135 grader. Gjennomsnittet må vektes i forhold til hvor mange piksler som er med ved beregningen av GLCM for hver av de fire retningene.

GLCM henter indirekte ut teksturinformatjon fra bildet. For at man skal kunne karakterisere bildeteksturen, er man nødt til å beregne egenskaper direkte fra matrisen. Disse egenskapene vil indirekte beskrive teksturen.

Mange klassiske GLCM egenskaper kan ses på som en vektet sum av verdiene til GLCM-elementene [46] [26] [10]. Vekten til hvert element blir bestemt av en gitt vektfunksjon $V(i,j)$. Man tilegner seg forskjellig informasjon om teksturen ved å endre vektfunksjonen. Disse vektfunksjonene kan deles inn

i to grupper. I den første gruppen blir vektfunksjonen basert på verdien til GLCM elementet. Et eksempel på en egenskap som hører til i denne gruppen er entropi [26] [46] som er gitt ved

$$Ent = - \sum_{i=1}^G \sum_{j=1}^G P(i, j) \log P(i, j), P(i, j) > 0$$

Her er $P(i, j)$ et element i GLCM, og vektfunksjonen er gitt ved $V(i, j) = -\log P(i, j)$. Entropien måler homogeniteten til et bilde. Hvis bildet bare har én gråtone vil entropien bli 0, mens i et bilde med mange forskjellige gråtoner vil entropien bli høy. For flere eksempler på egenskaper i denne gruppen se [26].

I den andre gruppen er vektfunksjonen basert på posisjonen i GLCM. Et eksempel på en egenskap som hører til i denne gruppen er ”inertia” [10] som er gitt ved

$$Inertia = \sum_{i=1}^G \sum_{j=1}^G (i - j)^2 P(i, j)$$

Her er $P(i, j)$ et element i GLCM, og vektfunksjonen er gitt ved $V(i, j) = (i - j)^2$. Inertia gir høye verdier for bilder med høy lokal kontrast.

Adaptiv egenskapsuttrekking

Man kan gjøre GLCM egenskaper adaptive [46] ved å kombinere klassiske GLCM metoder med et gitt antall frie parametere. Med adaptive egenskaper menes det at vektfunksjonen tilpasser seg bildematerialet mens man studerer det. De adaptive egenskapene vil hente ut informasjon fra GLCM på plasser hvor det er faktiske forskjeller mellom to klasser. Til sammenlikning henter de klassiske GLCM egenskapene ut informasjon fra matrisen på forhåndsbestemte plasser ved å vektlegge verdier eller posisjoner. Man må beregne tilhørende vektfunksjoner for å kunne trekke ut tilpasningsdyktige egenskaper. Disse vektfunksjonene er basert på enten en diskrimineringsmatrise [64] eller en klasseavstandsmatrise [44]. Vektfunksjonene blir beregnet ut ifra hele læringssettet av bilder fra to forskjellige klasser. De nevnte matrisene vil inneholde informasjon om hvor i GLCM man kan finne store klasseforskjeller. Det er verdt å merke seg at denne teknikken også kan benyttes på GLRLM, CGLRLM og GLEM som blir gjennomgått i henholdsvis 2.1.3, 2.1.4 og 2.1.5.

2.1.2 Gråtonenivå Sum- og Differansehistogram

Som et alternativ til GLCM kan man beregne de tilhørende gråtonenivå sum- og differansehistogrammene [46]. Denne tilnærmingen er mye mindre regnekrevende enn utregningen av GLCM. Som ved utregning av GLCM har man en gitt distanse, d , og en gitt retning θ . For hvert pikselpar med distanse d og retning θ blir summen og differansen mellom de aktuelle pikslene beregnet, og histogrammene blir oppdatert. Begge histogrammene blir normalisert etter at hele bildet er gjennomgått. I [63] introduserte Unser

et sett av sum- og differansehistogram egenskaper. Det viste seg at 9 av de originale GLCM egenskapene [26] kunne bli beregnet eksakt fra sum- og differansehistogrammene. Resultatene Unser kom frem til viste at sum- og differansehistogrammene ga omtrent like god klassifisering som den vanlige GLCM fremgangsmåten.

2.1.3 Gråtone Løpelengde-Matrise (GLRLM)

Et eksempel på en høyere ordens statistisk teksturegenskapmetode er Gråtone Løpelengde-Matrise, ”Gray-Level Run Length Matrix”, GLRLM [21] [9]. Piksler med samme gråtone liggende etter hverandre i en gitt retning er et gråtoneløp. Løpelengden er antall piksler i dette løpet, og løpelengdeverdien er antall forekomster av en gitt løpelengde. Nielsen et al. [46] har utformet en normalisert utgave av GLRLM, hvor elementene i matrisen er gitt ved

$$P(i, j | \theta) = \frac{1}{S} p(i, j | \theta), (i, j) \in (1, 1), \dots, (G, R)$$

Her er i gråtone, j er løpelengde, θ er en gitt retning, S er det totale antall løp i bildet og R er det lengste løpet. Når man bruker GLRLM er det vanlig å redusere antall gråtoner ved rekvantisering før beregningen av GLRLM. Man kan også benytte logaritmisk oppdeling av løpelengde-aksen for å fordele forekomstene bedre, siden det ofte blir en overhyppighet av korte løpelengder.

2.1.4 Samforekomst av Gråtone Løpelengde-Matrise (CGLRLM)

”Cooccurrence of Gray-Level Run Length Matrix”, CGLRLM [5] [46], er et alternativ til GLRLM og GLCM. CGLRLM er en fire-dimensjonal normalisert matrise $P(i, j, k, l)$, som inneholder estimert sannsynlighet for samforekomsten av to uavhengige løp gitt ved (gråtone, løpelengde) = (i, j) og (k, l) . De beste resultatene oppnås ved at de to uavhengige løpene (i, j) og (k, l) er to naboløp. Ved å følge Unsers [63] tankegang, kan man se på to uavhengige løp (i, j) og (k, l) som to tilfeldige variabler med samme varians. På den måten kan man erstatte den firedimensjonale CGLRLM-matrisen med tilhørende sum- og differanse-løpelengdematriser. Summen og differansen mellom de to gråtonene og de to løpelengdene blir beregnet for alle naboløp, og de to matrisene blir oppdatert i henhold til dette. Til slutt normaliseres begge matrisene. I forhold til GLCM og GLRLM gir CGLRLM mer informasjon om forholdet mellom gråtoneløpene. Det har imidlertid vist seg at høyere antall histogramelementer medfører at CGLRLM-egenskapene kan bli mindre robuste enn egenskapene til GLCM og GLRLM. I tillegg har CGLRLM vist seg å være mer sensitiv til støy i forhold til GLRLM, men dette problemet kan imidlertid løses ved å redusere antall gråtoner i bildet [46].

2.1.5 Gråtone Entropi-Matrise (GLEM)

Yogesani et al. [68] utviklet Gråtone Entropi-Matrise metoden, ”Gray-level Entropy Matrix” (GLEM), som baserer seg på høyere ordens statistikk. Her lar man et element i matrisen være gitt ved $P(i, j|w)$ hvilket tilsvarer sannsynligheten for entropiverdi j innenfor et vindu med størrelse $w \times w$. Verdien i representerer gråtoneverdien til pikselen lokalisert i sentrum av vinduet. Entropien er definert ved

$$E_j = - \sum_{z=1}^G P(z) \log P(z), \quad P(z) > 0$$

Her er $P(z)$ estimert sannsynlighet for forekomsten av gråtone z , i et $w \times w$ stort vindu. Det er verdt å merke seg at vindusstørrelse og antall gråtoner i bildet vil ha en innvirkning på hvor robust den estimerte gråtonefordelingen vil være. Som det ble diskutert over i forbindelse med entropiegenskapene til GLCM vil homogene strukturer gi lave entropiverdier, og inhomogene strukturer vil gi høye verdier.

2.1.6 Laws’ tekstur-energimålere

Laws’ tekstur-energimålere er basert på konvolusjon. Som navnet tilsier måles den lokale tekstur-energien. Dette gjøres ved å benytte en konvolusjonsmaske over bildet, som er vektet i forhold til det man ønsker å måle i den lokale teksten. Ofte kan denne masken være en binomisk vektet array, som blant annet kan forsterke forekomster av kanter, punkter eller komponenter med høye frekvenser. Det er heller ikke uvanlig å benytte større masker. Disse kan oppnås ved å konvolvare mindre masker med hverandre. Her er et par eksempler på konvolvering av mindre masker, for å få større masker:

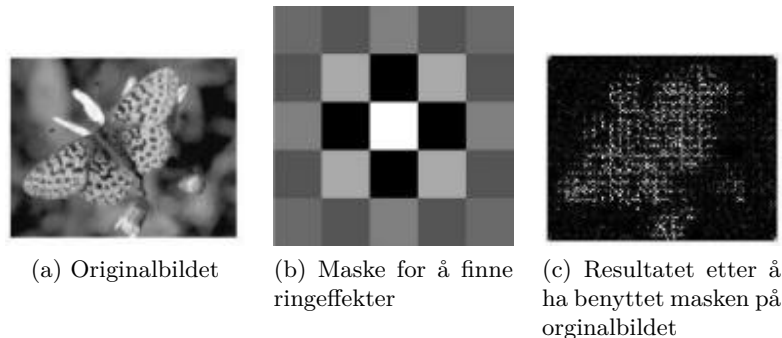
$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -4 & 6 & -4 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ -4 \\ 6 \\ -4 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \\ -4 & 16 & -24 & 16 & -4 \\ 6 & -24 & 36 & -24 & 6 \\ -4 & 16 & -24 & 16 & -4 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}$$

kan brukes til å finne ringeffekter. Et eksempel på resultat etter bruk av sistnevnte kan ses i figur 2.2. Harwood et al. [27] introduserer fire relaterte målinger basert på senter-symmetrisk auto-korrelasjon, SAC, SRAC, SCOV og SVR, som kan ses på som en generalisering av Laws’ tekstur-energimålere. Disse er gitt ved

$$SAC = \frac{\frac{1}{4} \sum_i^4 (x_i - \mu)(x'_i - \mu)}{\sigma^2}$$

$$SRAC = 1 - \frac{12 \{ \sum_i^4 (r_i - r'_i)^2 + T_x \}}{m^3 - m}$$



Figur 2.2: Eksempel på bruk av en av Laws' tekstur-energimålere. Bildene er hentet fra [51].

$$SCOV = \frac{1}{4} \sum_i^4 (x_i - \mu)(x'_i - \mu)$$

$$SVR = \frac{\frac{1}{16} \sum_i^4 (x_i - x'_i)^2}{\frac{1}{16} \sum_i^4 (x_i + x'_i)^2 - \mu}$$

hvor x_i og x'_i er to piksler i det digitale bildet som står symmetrisk overfor hverandre i forhold til et sentrumpiksel, μ er forventningsverdi og σ er standardavvik til gråtonene i det lokale nabolaget. Alle energimålerne er rotasjonsinvariante og robuste, og alle bortsett fra SCOV er lokalt gråtoneinvariante. Disse ser på et lokalt nabolag, for eksempel et med størrelse 3×3 , men med unntak av senterpiksel. På den måten får man i dette eksempelet fire senter-symmetriske par med piksler.

2.1.7 Lokalt Binært Mønster (LBP)

Det kan i noen tilfeller være fordelaktig å ha en gråtone-invariant operator for analyse av tekstonelementer, for eksempel for å se vekk fra lysvariasjon i et bilde eller i en bildeserie, noe GLCM, GLRLM, CGLRLM og GLEM ikke gjør. He og Wang [28] introduserte en opereror som baserer seg på lokal teksturinformasjon i et digitalt bilde. I likhet med målingene basert på senter-symmetrisk auto-korrelasjon med Laws' operatorer blir det også her sett på et lokalt nabolag. He og Wang ser på et 3×3 piksler stort nabolag $V = \{V_0, V_1, \dots, V_8\}$, hvor V_0 er gråtoneverdien til senterpiksel i nabolaget og $V_i \forall i \in [1, \dots, 8]$ er gråtoneverdiene til naboene rundt. De definerer så den tilsvarende teksturenheten ved $TU = \{E_1, E_2, \dots, E_8\}$ hvor

$$E_i = \begin{cases} 0, & \text{hvis } V_i < V_0 \\ 1, & \text{hvis } V_i = V_0 \\ 2, & \text{hvis } V_i > V_0 \end{cases}$$

He og Wangs operator er i likhet med SAC, SRAC og SVR gråtone-invariant, men ikke rotasjonsinvariant. På denne måten vil TU kunne ha $3^8 = 6561$ forskjellige kombinasjoner av dette tre-nivå mønsteret. Her lar de

en teksturenhet bli merket med nummeret gitt ved formelen

$$N_{TU} = \sum_{i=1}^8 E_i \times 3^{i-1} \quad (2.1)$$

I [61] presenterer Ojala et al. en to-nivå versjon av metoden til He og Wang hvor de lar alle naboer med gråtoneverdi større eller lik sentrumspikselet bli 1 og resten 0. På denne måten vil en teksturenhet bli et lokalt binært mønster ("Local Binary Pattern") [61] [54] [57] [4] [39] og det vil kun være $2^8 = 256$ forskjellige teksturenheter i stedet for 6561. Her blir en teksturenhet merket ved

$$N_{TU} = \sum_{i=1}^8 E_i \times 2^{i-1} \quad (2.2)$$

I [54] presenterer Ojala et al. en generell form for Lokalt Binært Mønster, $LBP_{P,R}$, hvor P er antall naboer man ser på, mens R er radiusen til "sirkelen" alle naboene ligger på. Her er det verdt å merke seg at hvis man har en radius $R = 1$, kan det maksimalt være $P = 8$ naboer. På denne måten begrenser man hva P kan være hvis R er satt, og motsatt. Tenker man på det digitale bildet som et koordinatsystem bestående av to dimensjoner, og lar sentrumspikselet, g_c , være origo $(0,0)$, er koordinatene til et element $g_p, \forall p \in [0, \dots, P-1]$ i nabosettet $[g_0, g_1, \dots, g_{P-1}]$ gitt ved

$$(-R \sin(2\pi p/P), R \cos(2\pi p/P))$$

Når man har plukket ut et nabolag gjenstår det kun å trekke ifra gråtoneverdien til senterpikselen fra hver piksel i nabolaget, og deretter avgjøre hvorvidt en plass i sekvensen skal være 0 eller 1, slik som før. Den generelle formen som blir presentert i [54] er gitt ved $[s(g_0 - g_c), s(g_1 - g_c), \dots, s(g_{P-1} - g_c)]$, hvor g_c er gråtoneverdien til senterpikselen og

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Denne sekvensen med naboer kan dermed gjøres om til et binært tall. Har man for eksempel $[0, 0, 1, 0, 0, 1, 0, 1]$ lar man dette tilsvare det binære tallet 00100101, som igjen er lik 37 i titallsystemet. På denne måten får man unike $LBP_{P,R}$ tall, også kalt "Texture Unit" (TU), som forteller noe om det lokale området. Altså er den generelle formen til et tekstonelement, som formel (2.2) er et eksempel på, gitt ved

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p$$

Ettersom man ser på et sett med P naboer, vil man få en P -bits streng. Dette vil dermed medføre at $LBP_{P,R}$ vil kunne gi ut 2^P forskjellige verdier. Dette antallet kan man redusere ved å gjøre operatoren rotasjonsinvariant. For å oppnå dette foreslår Ojala et al. [54] å utføre sykliske rotasjoner av den binære sekvensen og dermed finne bit-sekvensen som representerer det laves-te tallet. På denne måten vil for eksempel $LBP_{8,1}^{ri}$ kunne returnere opptil 36

forskjellige verdier. Her er den hevede ri tatt med for å vise at operatoren er rotasjonsinvariant. Se figur 2.3 for å se alle de rotasjonsinvariante mønstrene til $LBP_{8,1}^{ri}$.

Isteden for å lete etter den minste verdien en bitstreng kan ha for hver LBP, foreslår Ojala et al. [54] at man kan benytte en lineær oppslagstabell (LUT - "Look-Up-Table"). Albregtsen [4] forslår videre at man kan ha en 16×16 stor LUT, i tilfellet med 8 naboer, hvor man lar kolonne- og radindeksene være henholdsvis de 4 første og siste bitene i bit-sekvensen.

På denne måten vil et dataprogram bruke mindre CPU-tid for å beregne LBP. Generelt kan man benytte $2^{P/2} \times 2^{P/2}$ store oppslagstabeller, men dette kan medføre høyt minneforbruk. En LUT for $LBP_{8,1}$ er å finne i figur 2.4.

LBP-operatoren kan gjøres speilingsinvariant [4] ved å speile mønsteret om den horisontale, vertikale, diagonale og den bi-diagonale aksene. Ved å gjøre dette oppnår man fremdeles et mønster som tilsvarer et tall mellom 0 og 255. Ved å anvende dette på de 36 mønstrene i 2.3 kan man se bort ifra ytterligere 6 mønstre. Se figur 2.5 for eksempel. De mønstrene som blir endret er 13 til 11, 25 til 19, 29 til 23, 53 til 43, 59 til 55 og 61 til 47. Vi kan dermed forenkle figur 2.3 til å bestå av 30 mønstre, se figur 2.6. Ved speiling vil ikke antall 0/1 overganger endres i mønstrene. Dette fører til at histogrammer som viser mønstrene gruppert etter antall overganger vil bli uforandret, til tross for at noen mønstre slås sammen.

Som påpekt av både Ojala et al. [54] og Albregtsen [4] har det vist seg i praksis at det er noen fundamentale mønstre som forekommer hyppigere enn resten. Felles for disse mønstrene er at de har få overganger mellom 0 og 1. Dette kommer av at teksturer i lokale områder har en tendens til å være enten et flatt område, myke variasjoner fra mørk til lys eller motsatt, eller mørke/lyse prikker i bildet. Dette tilsvarer de 9 øverste mønstrene i figur 2.3. Ettersom forekomsten av "ikke-uniforme" mønstre er såpass lav, kan ikke forekomsten av disse bli estimert nøyaktig i et histogram. Av denne grunnen foreslår Ojala et al. [54] en LBP-operator som klassifiserer alle "ikke-uniforme" mønstre som ett mønster, mens alle "uniforme" blir behandlet som før. Her definerer de et "uniformt" mønster til å ha maks to 0/1 overganger. Altså blir den nye LBP-operatoren seende slik ut:

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{hvis } U(LBP_{P,R}) \leq 2 \\ P + 1 & \text{ellers} \end{cases}$$

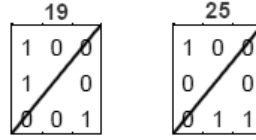
hvor

$$U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|$$

Denne operatoren er nå veldig god til å finne romlige mønstre, men ettersom den kun tar hensyn til hvilket fortegn $n_p = g_p - g_c$ har, hvor g_p er en nabopiksel og g_c er sentrumpiksel, forkaster man informasjonen om

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	0	1	1	3	1	5	3	7	1	9	5	11	3	13	7	15
0001	1	17	9	19	5	21	11	23	3	25	13	27	7	29	15	31
0010	1	9	17	25	9	37	19	39	5	37	21	43	11	45	23	47
0011	3	19	25	51	13	53	27	55	7	39	29	59	15	61	31	63
0100	1	5	9	13	17	21	25	29	9	37	37	45	19	53	39	61
0101	5	21	37	53	21	85	43	87	11	43	45	91	23	87	47	95
0110	3	11	19	27	25	43	51	59	13	45	53	91	27	91	55	111
0111	7	23	39	55	29	87	59	119	15	47	61	111	31	95	63	127
1000	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
1001	9	25	37	39	37	43	45	47	19	51	53	55	39	59	61	63
1010	5	13	21	29	37	45	53	61	21	53	85	87	43	91	87	95
1011	11	27	43	59	45	91	91	111	23	55	87	119	47	111	95	127
1100	3	7	11	15	19	23	27	31	25	39	43	47	51	55	59	63
1101	13	29	45	61	53	87	91	95	27	59	91	111	55	119	111	127
1110	7	15	23	31	39	47	55	63	29	61	87	95	59	111	119	127
1111	15	31	47	63	61	95	111	127	31	63	95	127	63	127	127	255

Figur 2.4: Look-Up-Table (LUT) for LBP. Tabellen er farget etter antall 0/1-overganger og som man kan se, er de mest homogene bit-sekvensene (f.eks. 1 og 3) plassert symmetrisk i kantene, mens for eksempel den mest inhomogene sekvensen, 85, er plassert på to symmetriske plasser langs diagonalen. Bildet er hentet fra [4]



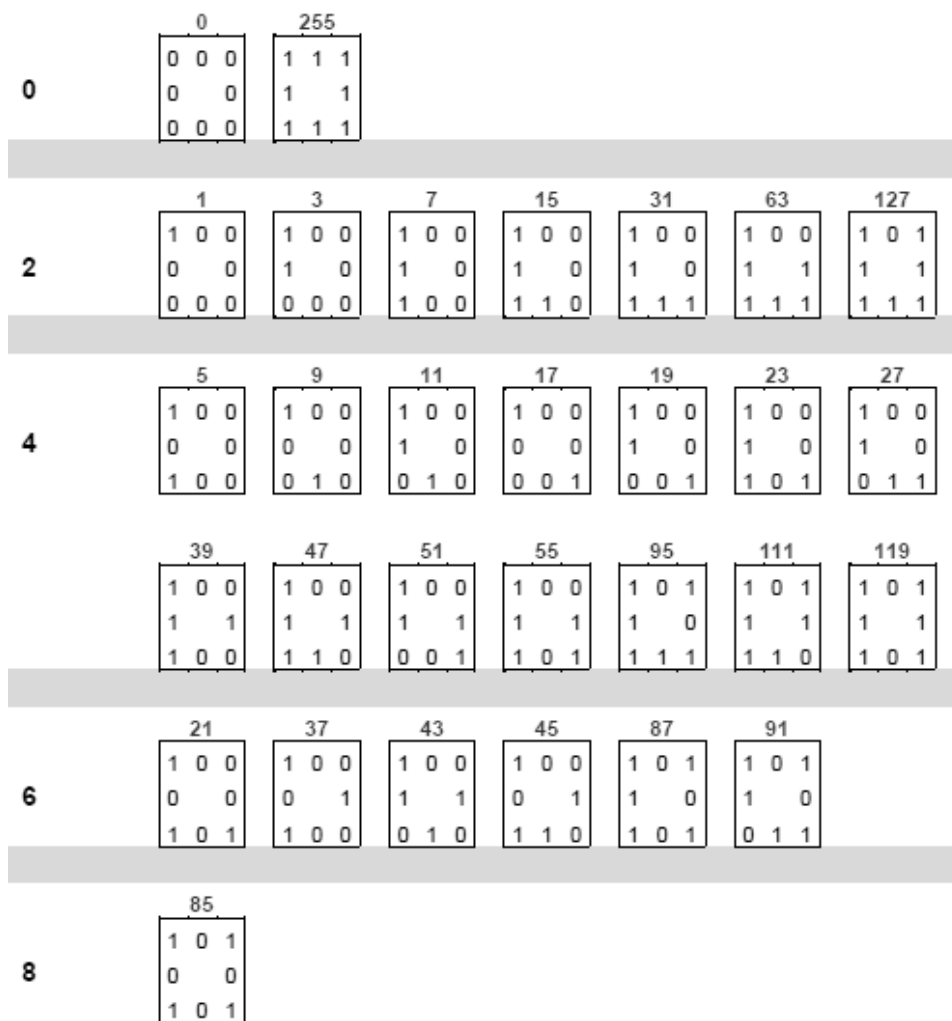
Figur 2.5: Speiler 19 om diagonalen og får 25. Bildet er hentet fra [4], men med små modifikasjoner.

kontrasten. Hvis gråtone-invarians ikke er krevet, foreslår Ojala et al. [54] å benytte en rotasjonsinvariant måling av den lokale variansen. Formelen for dette er gitt ved:

$$VAR_{P,R} = \frac{1}{P} \sum_{p=0}^{P-1} (g_p - \mu)^2, \quad \mu = \frac{1}{P} \sum_{p=0}^{P-1} g_p$$

Som man kan se er denne operatoren tilpasset i forhold til hvor mange naboer man ønsker å se på samt radius, akkurat som $LBP_{P,R}$, og på den måten kan disse slås sammen for å benyttes på det samme området. En annen ting som er verdt å merke seg, er at disse operatorene utfyller hverandre, og av den grunn kan det forventes at de vil gjøre en god jobb sammen. Man kan derfor benytte den samlede operatoren $LBP_{P,R}^{riu2}/VAR_{P,R}$. På tilsvarende måte er det også mulig å kombinere LBP med andre egenskaper som for eksempel entropi.

I [54] ble $LBP_{P,R}^{riu2}$ og $LBP_{P,R}^{riu2}/VAR_{P,R}$ testet på 16 teksturklasser fra Brodatz albumet [7]. Her ble klassifikatoren trent på en rotasjonsvinkel og testet på andre rotasjonsvinkler. Resultatet viste seg å bli veldig bra. Når man ser bort fra kombinasjon av operatører, gjorde $LBP_{24,3}^{riu2}$ det best etterfulgt $LBP_{16,2}^{riu2}$ og $LBP_{8,1}^{riu2}$. Resultatet er ikke overraskende ettersom høyere P gjør at operatoren tar hensyn til flere vinkler. I tillegg viste det seg at i de aller fleste tilfellene ble resultatet bedre for $LBP_{P,R}^{riu2}/VAR_{P,R}$, med



Figur 2.6: 30 unike rotasjons- og speilingsinvariante binære mønstre. Bildet er hentet fra [4] Mønstrene er gruppert etter antall overganger fra 0 til 1 eller 1 til 0 (se venstre kolonne).

unntak for $P = 24$ og $R = 3$, hvor resultatet ble litt dårligere. Det viste seg også at det $LBP_{8,1}$ lærte fra vinklene som går opp i 45 ikke generaliserte noe godt i forhold til andre vinkler. De aller beste resultatene ble oppnådd i kombinasjonen $LBP_{P,R}^{riu2}/VAR_{P,R}$ med $8, 1 + 24, 3$ og $8, 1 + 16, 2 + 24, 3$, mens $LBP_{16,2}^{riu2}/VAR_{16,2}$ fulgte rett etter. Videre ble de samme operatorene og kombinasjonene testet på teksturer fra Outex [52]. Også her viste det seg at $LBP_{24,3}$ gjorde det best, mens $LBP_{8,1}$ gjorde det dårligst blant de enkeltstående LBP-operatorene. Kombinasjonene $LBP_{P,R}^{riu2}/VAR_{P,R}$ gjorde det totalt sett bedre for alle P, R i forhold til $LBP_{P,R}$ med tilsvarende P, R .

Pietikäinen et al. testet en rotasjonsinvariant versjon av LBP-operatoren gitt i formel (2.2), som de her kalte LBPROT [56]. Denne ble testet opp mot andre rotasjonsinvariante metoder som SAC, SRAC, SCOV og SVR [27], og er definert likt som over. I tillegg ble også "circular symmetric autoregressive random field model" (CSAR) [34], varians (VAR) og DIFF4 brukt. CSAR er gitt ved

$$y(s) = \alpha \sum_{r \in N_c} g_r y(s \oplus r) + \sqrt{\beta} v(s)$$

hvor s er pikselkoordinatene til en piksel i et $M \times M$ stort digitalt bilde, $\{y(s), s \in \omega, \omega = (0 \leq s_1, s_2 \leq M-1)\}$ er et sett med gråtoneverdier, N_c er nabolagssettet med piksler, $v(s)$ er en korrelert sekvens, α og β er to koeffisienter og måler henholdsvis en bestemt isotropisk egenskap og røffheten i bildet. \oplus står her for modulo M addisjon. Varians er gitt ved

$$VAR = \frac{1}{8} \sum_i^4 (x_i^2 + x_i'^2) - \mu^2$$

hvor x_i og x_i' er et sentersymmetrisk par og μ er den lokale forventningsverdien. DIFF4 måler den absolutte differansen mellom nabopiksler i de fire hovedretningene ($\theta = 0, 45, 90$ og 135 grader). Det er brukt 15 forholdsvis like klasser av Brodatz teksturer [7]. Her ble det også benyttet to forskjellige vindusstørrelser for å måle effekten vindusstørrelsen har på de forskjellige metodene. Med en vindusstørrelse på 64×64 viste det seg at de gråtoneinvariante metodene (LBPROT, SVR, SAC, SRAC) gjorde det dårligst, mens VAR og SCOV gjorde det best. Med vindusstørrelse på 32×32 blir feilene større, hvilket er forståelig ettersom kun en fjerdedel av pikseldataen blir betraktet. Ved kombinasjon av metoder viser det seg, i likhet med eksperimentene utført i [54], at LBPROT sammen med VAR gjør det veldig bra, og er blant de beste kombinasjonene med begge vindusstørrelsene. Dersom man ser på de ulike kombinasjonene med LBPROT, kan man legge merke til at kombinasjonene med de ikke-gråtoneinvariante operatorene gjør det bedre enn kombinasjonene med de gråtoneinvariante.

LBP baserer seg på lokale områder i bildet, og er konstruert slik at operatoren er gråtoneinvariant. Denne egenskapen kommer godt med i situasjoner hvor man ønsker å se bort fra variasjoner i bilder på grunn av ulik lyssetting. Hvis gråtoneinvarians ikke er et krav, er det mulig å kombinere

LBP med operatorer som ikke er gråtoneinvariante. Resultatene over viser at rotasjonsinvariant LBP kombinert med for eksempel varians gir en svært kraftig operator. Det er også mulig å kombinere LBP med andre operatorer som i likhet med LBP er gråtoneinvariante. Resultatene over gir imidlertid inntrykk av at slike kombinasjoner ikke nødvendigvis er like kraftige.

Som vi har sett er det mulig å gjøre LBP rotasjonsinvariant, og på den måten gjøre operatoren mer robust i forhold til ulike rotasjoner av for eksempel tekstonelementer i bilder. På denne måten vil man også redusere antall utverdier, og dermed gjøre det enklere å klassifisere ulike områder i bildet. Man kan også redusere antall utverdier ved å gjøre operatoren speilinginvariant. Det er imidlertid verdt å merke seg at man i praksis kun ender opp med en tilnærming til rotasjon- og speilinginvarians og at denne tilnærmingen blir bedre desto flere nabo piksler man ser på i en sirkel rundt senterpiksel. Dette så vi et eksempel på i resultatene over hvor $LBP_{8,1}$ ga dårligere resultater på teksturer med vinkler som ikke gikk opp i 45 grader, i forhold til $LBP_{16,2}$ og $LBP_{24,3}$. Dette kommer av at naboene er jevnt plassert i en sirkel rundt senterpiksel, og flere naboer medfører mindre vinkel mellom hver nabo. En ulempe med å øke antall naboer er at naboene vil befinne seg lenger unna selve senterpiksel. Ved å legge sammen nabolag med ulik radius og antall naboer, kan man oppnå en rotasjonsinvariant LBP-operator som tar hensyn til flere vinkler samtidig som man beholder informasjon fra naboer nær senterpiksel. Over så vi at slike kombinasjoner ga gode resultater. Ulempen med dette er at man vil ende opp med å se på mange piksler rundt et senterpiksel og bit-sekvenser kan bli veldig lange. Noe som igjen medfører at operatoren blir mer regnekrevende. Fra dette kan man se at valg av LBP-operator avhenger av problemet man har for hånd, både med tanke på størrelse på bildet, om man kan se bort ifra gråtone-, rotasjons- og speilinginvarians og hvor mye tid man har til rådighet ved kjøring av operatoren.

2.2 Distansemålere

Dersom man har sannsynlighetsfordelingen til 2 klasser, representert ved 2 histogrammer P og Q, kan man ønske å finne et distansemål på hvor langt det er fra en pasient til hver av fordelingene. På denne måten får man et mål man kan legge til grunn for hvilken gruppe en pasient hører til. Det finnes en rekke ulike distansemålere og i denne seksjonen tar vi for oss flere av disse.

2.2.1 Mahalanobis distanse

Mahalanobis distanse [37] [41] er et enkelt mål på distansen mellom to statistiske objekter, og kan blant annet bli brukt til å måle likheten mellom diverse sannsynlighetsfordelinger. En ting man må være oppmerksom på når man finner Mahalanobis distanse, er dens antakelse om at de statistiske objektene er normalfordelte. Skulle det være tilfellet at de statistiske objektene ikke er normalfordelte, blir det straks mer fordelaktig å benytte andre distansemålere som ikke antar normalfordeling ettersom Mahalanobis

distansene vil kunne bli svært unøyaktige.

Generelt er Mahalanobis distansen definert som

$$J_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)}$$

hvor $\mathbf{x} = (x_1, x_2, \dots, x_N)^T$ er en gruppe med målinger, med forventningsverdier $\mu = (\mu_1, \mu_2, \dots, \mu_N)^T$ og Σ er kovariansmatrisen. For å finne forskjellen mellom to klasser, K_0 og K_1 , kan man måle og summere opp forskjellen mellom disse to klassene [41]. En måte å gjøre dette på er å bruke den kvadrerte Mahalanobis distansen

$$J_M^2 = (\mu_1 - \mu_0)^T \Sigma^{-1} (\mu_1 - \mu_0)$$

hvor μ_0 og μ_1 er vektorer som inneholder forventningsverdien til hver av egenskapene for henholdsvis K_0 og K_1 , og Σ er kovariansmatrisen gitt ved

$$\Sigma = N_0 \Sigma_0 + N_1 \Sigma_1$$

hvor N_0 og N_1 er sannsynligheten for henholdsvis K_0 og K_1 . Dersom man kun har en egenskap for hver klasse og antar at $N_0 = N_1 = \frac{1}{2}$, vil man ende opp med

$$\Sigma = \frac{1}{2} \sigma_0^2 + \frac{1}{2} \sigma_1^2$$

Dette kan man så sette inn i J_M^2 , noe som gjør at man ender opp med

$$J_2 = (\mu_1 - \mu_0) \left(\frac{1}{2} \sigma_0^2 + \frac{1}{2} \sigma_1^2 \right)^{-1} (\mu_1 - \mu_0)$$

$$J_2 = (\mu_1 - \mu_0)^2 \left(\frac{1}{2} (\sigma_0^2 + \sigma_1^2) \right)^{-1}$$

$$J_2 = \frac{2(\mu_1 - \mu_0)^2}{\sigma_0^2 + \sigma_1^2}$$

hvor μ_0 og σ_0 er henholdsvis forventningsverdien og variansen til K_0 , og tilsvarende for K_1 . Siden det å ta kvadratroten kun er en skalering av et gitt tall, vil man også kunne bruke $\sqrt{J_2}$. Noen ganger ønsker man i tillegg et fortegn på distansene. Dette kan man få ved

$$J_3 = f J_2$$

hvor

$$f = \begin{cases} 1 & \text{hvis } \mu_1 \geq \mu_0 \\ -1 & \text{ellers} \end{cases}$$

2.2.2 Bhattacharyya

I tillegg til å anta normalfordeling legger Mahalanobis distanse stor vekt på forventningsverdien i forhold til ulikheter i formen til de statistiske objektene. En distansemåler som ikke antar normalfordeling er Bhattacharyya distanse [6] [24]. Dette distansemålet vektlegger formen til de statistiske objektene i større grad enn Mahalanobis, hvilket igjen betyr at vektleggingen av forventningsverdiene blir lavere. Generelt er Bhattacharyya distansen gitt ved

$$J(\omega_1, \omega_2) = -\ln \int \sqrt{p(\mathbf{x}|\omega_1)p(\mathbf{x}|\omega_2)} d\mathbf{x}$$

hvor $p(\mathbf{x}|\omega_i)$ er sannsynligheten for at egenskapsvektoren \mathbf{x} tilhører klasse ω_i . En annen fordel med denne distansemåleren er at man kan estimere en øvre og nedre grense for feil, ε , i klassifikasjonen. Disse grensene er gitt ved

$$\frac{1}{2}[1 - (1 - 4P(\omega_1)P(\omega_2)e^{-2J})^{1/2}] \leq \varepsilon \leq \sqrt{P(\omega_1)P(\omega_2)}e^{-J}$$

hvor $P(\omega_i)$ er sannsynligheten for klasse ω_i . Ved å beregne denne kan man finne ut i hvilken grad man kan stole på distansen man har regnet ut, som igjen vil ha en innvirkning på i hvilken grad man kan stole på klassifikasjonen som baserer seg på denne distansen.

For å skille mellom sannsynlighetsfordelinger kan man også benytte seg av Bhattacharyya koeffisienten [16]. Denne tilpasningen er gitt ved

$$\rho(p, p') = \sum_{i=1}^N \sqrt{p(i)p'(i)}$$

Her representerer p og p' to sannsynlighetsfordelinger og $\sum_{i=1}^N p(i) = \sum_{i=1}^N p'(i) = 1$. Denne koeffisienten har imidlertid ikke en metrisk struktur [20], noe som gjør at sammenlikninger kan bli vanskeligere. For å løse dette problemet foreslår Fukunaga [20] en modifikasjon av Bhattacharyya koeffisienten slik at distansen oppfyller aksiomene som ligger til grunn for metriske distanser. Denne modifikasjonen er gitt ved

$$d(p, p') = \sqrt{1 - \rho(p, p')}$$

Selv om Bhattacharyya ikke antar noen bestemt fordeling, kan det i flere tilfeller hende at en bestemt fordeling er til stede, som for eksempel normalfordeling. I disse tilfellene kan en distansemåler som antar denne fordelingen oppnå et bedre resultat. Det finnes flere ulike modifikasjoner av Bhattacharyya som er tilpasset ulike sannsynlighetsfordelinger, men her kommer vi kun til å ta for oss tilpasning for normalfordelingen. Denne er gitt ved

$$\frac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \left(\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2} \right)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2} \ln \frac{|\frac{1}{2}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)|}{|\boldsymbol{\Sigma}_1|^{1/2}|\boldsymbol{\Sigma}_2|^{1/2}}$$

hvor $\boldsymbol{\mu}_i$ og $\boldsymbol{\Sigma}_i$ er henholdsvis forventningsverdivektoren og kovariansmatrisen til klasse ω_i . Ved bruk av denne formelen på normalfordelte sannsynlighetsfordelinger kan man forvente seg et mer nøyaktig estimat på distansen, enn ved bruk av generell Battacharyya. Skulle det imidlertid vise seg at sannsynlighetsfordelingene ikke er normalfordelte, bør man i stedet benytte seg av tilpasninger til de aktuelle sannsynlighetsfordelingene, enn akkurat denne modifikasjonen.

2.2.3 Chi-kvadratfordeling

I mange tilfeller vil fordelingene man skal måle avstander mellom ha en annen form enn formen til en normalfordeling. En annen typisk fordeling man kan komme borti er Chi-kvadratfordelingen [17]. For å måle hvor usannsynlig det er at den ene fordelingen er trukket fra populasjonen som er representert ved den andre fordelingen kan man i følge [35] bruke Chi-kvadrat signifikanstesten. Denne testen er gitt ved

$$\chi^2(p, p') = \sum_{i=1}^N \frac{(p(i) - p'(i))^2}{p'(i)}$$

hvor $p(i)$ og $p'(i)$ er to sannsynlighetsfordelinger.

2.2.4 "Earth Mover's Distance"

Et litt annet distanse mål, som ikke antar noen fordeling, er "Earth Mover's Distance" (EMD) [59]. Når man har to fordelinger, kan man se på den ene fordelingen som en jordhaug, mens den andre fordelingen kan bli sett på som et hull. EMD måler da det minste arbeidet som må til for å flytte jordhaugen ned i hullet. EMD er formelt definert på følgende måte: Ønsker å finne en flyt som minimerer den totale kostnaden gitt ved

$$WORK(P, Q, F) = \sum_{i=1}^m \sum_{j=1}^n d_{ij} f_{ij}$$

hvor $P = \{(\mathbf{p}_1, \omega_{\mathbf{p}_1}), \dots, (\mathbf{p}_m, \omega_{\mathbf{p}_m})\}$ og $Q = \{(\mathbf{q}_1, \omega_{\mathbf{q}_1}), \dots, (\mathbf{q}_n, \omega_{\mathbf{q}_n})\}$ er henholdsvis det første og det andre histogrammet med respektive m og n grupper. \mathbf{p}_i er gruppe-representanten og ω_i er vekten til gruppen, det tilsvarende gjelder for Q . $D = [d_{ij}]$ er en matrise som inneholder grunndistansene, hvor d_{ij} er grunndistansen mellom \mathbf{p}_i og \mathbf{q}_j . Flyten er definert som $\mathbf{F} = [f_{ij}]$, hvor f_{ij} er flyten mellom \mathbf{p}_i og \mathbf{q}_j . For å minimere kostnaden må man oppfylle følgende kriterier:

$$f_{ij} \geq 0 \quad 1 \leq i \leq m, 1 \leq j \leq n \quad (2.3)$$

$$\sum_{j=1}^n f_{ij} \leq \omega_{\mathbf{p}_i} \quad 1 \leq i \leq m \quad (2.4)$$

$$\sum_{i=1}^m f_{ij} \leq \omega_{\mathbf{q}_j} \quad 1 \leq j \leq n \quad (2.5)$$

$$\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min \left(\sum_{i=1}^m \omega_{\mathbf{p}_i}, \sum_{j=1}^n \omega_{\mathbf{q}_j} \right) \quad (2.6)$$

Når man har løst dette er EMD definert til å være arbeidet normalisert med den totale flyten:

$$EMD(P, Q) = \frac{\sum_{i=1}^m \sum_{j=1}^n d_{ij} f_{ij}}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}}$$

2.2.5 Andre distansemålere

I [8] står det beskrevet flere distansemålere. En av disse er Minkowski distansen, og er definert som

$$d_{L_p}(P, Q) = \left(\sum_i |P_i - Q_i|^p \right)^{1/p}$$

Minkowski distansen er en generalisering av Euklidsk distanse hvor $p = 2$, City Block distanse hvor $p = 1$, og Chebyshevs distanse hvor $p = \infty$.

To andre mye brukte distansemålere er Kullback-Leibler og Jeffreys divergens. Førstnevnte er gitt ved

$$d_{KL} = \sum_{i=1}^d P_i \ln\left(\frac{P_i}{Q_i}\right)$$

mens sistnevnte, som er en litt annen versjon av Kullback-Leibler, er gitt ved

$$d_J = \sum_{i=1}^d P_i \ln\left(\frac{2P_i}{Q_i + P_i}\right) - Q_i \ln\left(\frac{2Q_i}{Q_i + P_i}\right)$$

En tredje mulighet er å måle skjæringspunktet mellom de to histogrammene, denne metoden går under navnet Intersection og er definert på følgende måte

$$d_{non-IS} = 1 - S_{IS} = \frac{1}{2} \sum_{i=1}^d |P_i - Q_i|$$

$$S_{IS} = \sum_{i=1}^d \min(P_i, Q_i)$$

Her kan man legge merke til at d_{non-IS} tilsvarer City Block distanse delt på 2.

2.2.6 Oppsummering

Vi har nå sett på ulike måter å måle avstand på. Dersom dataene man skal måle avstanden mellom er normalfordelte, og det er forskjell i forventningsverdiene, vil Mahalanobis distanse være et godt valg av distansemål.

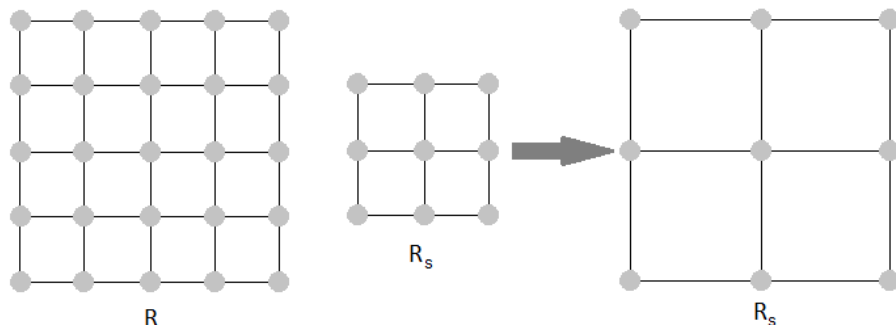
Men hvis det ikke er noen særlig forskjell mellom forventningverdiene kan man med fordel benytte seg av Gaussisk Bhattacharyya. Denne metoden tar mer hensyn til variansen i dataene. Dersom dataene man har ikke er normalfordelte, kan man enten benytte Bhattacharyya tilpasset en annen sannsynlighetsfordeling eller Chi-kvadrat signifikanstesten. Hvis man ikke ønsker å benytte en distansemåler som er tilpasset en bestemt fordeling, kan man benytte seg av den generelle Bhattacharyya distansen, eller eventuelt Bhattacharyya koeffisienten. Dette er spesielt aktuelt når det er vanskelig å beskrive sannsynlighetsfordelinger med en gitt modell. Man kan også benytte seg av "Earth Mover's Distance", men dette er en mye mer komplisert metode å benytte, så vi ser bort fra bruken av EMD, i denne oppgaven.

2.3 Nedskalering og bildepyramide

Når man skal analysere tekstur, er det i mange tilfeller vanskelig å avgjøre hvor stor andel av teksturobjektet man skal analysere om gangen. Dersom man analyserer et for stort område kan man risikere å sitte igjen med for generell informasjon om teksturen. Skulle man derimot analysere et for lokalt område vil ikke informasjonen man ender opp med være tilstrekkelig til å beskrive teksturen man er interessert i. Man kan i mange tilfeller anta hvor stort område man bør analysere av gangen, gjerne på grunnlag av hvor stor plass et teksturelement opptar før det gjentar seg. Ved analyse av cellekjerne-bilder er vi imidlertid ikke så heldige, blant annet fordi vi ikke kan si med sikkerhet hva slags tekstur som skiller cellekjerne-bilder med god og dårlig prognose i for eksempel prostatakreft. I stedet kan vi bare anta at slike områder finnes, enten ved at det kun forekommer i det ene prognostiske tilfellet, eller at det forekommer i begge, men med ulik hyppighet.

For å finne en tilnærming til hvor mye av cellekjernen man skal se på om gangen, når man skal analysere teksturen med LBP, har vi i utgangspunktet to veier å gå. Det første man kan gjøre er å teste med ulike størrelser på nabolaget, for eksempel ved å bruke LBP-operatorer som ser på alle piksler som ligger på en sirkel med ulik radius, slik som Ojala et al. foreslår i [54]. I tillegg kan LBP-operatorer med ulik radius slås sammen for å danne et bedre bilde av teksturen rundt en senterpiksel. En fordel med denne fremgangsmåten er at LBP-operatoren blir rotasjons-invariant til flere vinkler desto større nabolag en ser på. Med andre ord gir dette en bedre tilnærming til rotasjonsinvarians. Ulempen med denne metoden er at den er mer regnekrevende og vil medføre langt flere ulike verdier, noe som igjen vil påvirke kvantiseringsprosessen av mulige utverdier. Økning av nabolaget er allerede beskrevet i seksjon 2.1.7, så vi kommer ikke til å gå noe videre inn på dette her.

Den andre måten man kan analysere ulike andeler av teksturer på er ved å skalere ned bildet før man benytter en forhåndsbestemt LBP-operator. Denne fremgangsmåten tillater bruk av en mindre regnekrevende operator, ettersom man ikke trenger å se på flere piksler for å danne seg et bilde av



Figur 2.7: Viser et nett R med størrelse 5×5 og et nett R_s med størrelse 3×3 . Viser så hvordan R_s blir strukket for å oppta samme område som R .

den lokale teksturen. Ved bruk av denne metoden ender man imidlertid ikke opp med en like god tilnærming til rotasjonsinvarians i operatoren. I tillegg til dette er man, på grunn av nedskaleringen, nødt til å approksimere piksler under skaleringen. Dette medfører tap av både romlig og gråtone-basert informasjon, og bidrar til å gjøre endringer i den lokale teksturen. Hvor mye den lokale teksturen blir endret under nedskalering avhenger av hvor mye man skalerer bildet, samt hvilke teknikker man benytter seg av under skaleringen. Vi gjennomgår slike teknikker i seksjonen som omhandler nedskalering, før vi tar for oss en struktur kalt bildepyramide, som lar oss representere bilder i mer enn en oppløsning.

2.3.1 Nedskalering

Skalering av et bilde [23] går i korte trekk ut på å endre oppløsningen til bildet. Ved å øke oppløsningen til et bilde, sier vi at vi skalerer opp bildet, og tilsvarende skalerer vi ned bildet ved å redusere oppløsningen til bildet. Det er sistnevnte vi konsentrerer oss om i denne seksjonen.

Når man skal se for seg en nedskalering av et bilde med størrelse $M \times N$ piksler tenker vi oss at vi først lager et nett R med $M \times N$ punkter. Et punkt tilsvarer en piksel i bildet på den måten at gråtonene/fargene er like, og den romlige oppløsningen i nettet er lik den i bildet. Når man så skal skalere ned dette bildet til et $A \times B$ stort bilde, lager vi et nytt nett, R_s , bestående av $A \times B$ punkter. Disse punktene inneholder ingen gråtone-/fargeverdier, men vi antar at avstanden mellom punktene i R_s er lik avstanden mellom punktene i R . Vi strekker så ut R_s slik at dette nettet opptar tilsvarende område som R . En illustrasjon av dette er vist i figur 2.7. Etter at denne strekkingen er gjort legger vi R_s over R , og finner en måte å tilegne et punkt i R_s en verdi basert på nærliggende punkter i R . Det er flere måter å tilegne disse verdiene, og samlebegrepet for slike tilegninger kalles interpolasjon.

I [23] tar Gonzalez og Woods for seg tre populære interpolasjonsmetoder kalt nærmeste nabo, bilinear og bikubisk interpolasjon. Nærmeste nabo interpolasjon er den enkleste av disse, og går ut på å la et punkt i R_s få

samme verdi som det nærmeste punktet i R . Denne metoden har dessverre en tendens til å medføre uønskede effekter på det nye bildet, som for eksempel å gjøre tidligere rette kanter hakkete.

For å redusere slike uønskede effekter kan man, for eksempel, se på de 4 nærmeste naboene et punkt i R_s har i R . Fra disse 4 punktene finner vi en tilnærming til gråtonen punktet i R_s får. Denne fremgangsmåten kalles bilinear interpolasjon. Et eksempel på bruk av disse 4 naboene kan være

$$r_s(x, y) = r(\lfloor x \rfloor, \lfloor y \rfloor)(1 - \Delta x)(1 - \Delta y) + r(\lfloor x \rfloor + 1, \lfloor y \rfloor)\Delta x(1 - \Delta y) + r(\lfloor x \rfloor, \lfloor y \rfloor + 1)(1 - \Delta x)\Delta y + r(\lfloor x \rfloor + 1, \lfloor y \rfloor + 1)\Delta x\Delta y$$

hvor $r_s(x, y)$ er gråtonen til et punkt i R_s , r er gråtone i R og $\lfloor \cdot \rfloor$ er operatoren for å runde ned til nærmeste heltall og

$$\begin{aligned}\Delta x &= x - \lfloor x \rfloor, \\ \Delta y &= y - \lfloor y \rfloor,\end{aligned}$$

Som sagt medfører denne metoden bedre resultater enn nærmeste nabo interpolasjon, og dette gjør den uten å øke regnetiden i altfor stor grad.

Som vi har vært inne på tidligere har vi også bikubisk interpolasjon. I likhet med bilinear interpolasjon ser vi også her på flere naboer, men i stedet for å se på kun 4, ser vi nå på de 16 nærmeste naboene. En gråtone $r_s(x, y)$ i R_s kan bli funnet på tilsvarende måte som tidligere. Med andre ord blir

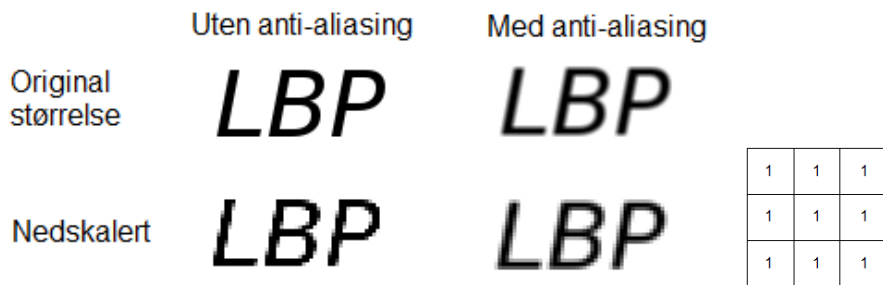
$$r_s(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 r(\lfloor x \rfloor - 1 + i, \lfloor y \rfloor - 1 + j) a_{ij}$$

hvor a_{ij} er en vekt som settes i forhold til hvor langt $(\lfloor x \rfloor - 1 + i, \lfloor y \rfloor - 1 + j)$ er fra (x, y) og

$$\sum_{i=0}^3 \sum_{j=0}^3 a_{ij} = 1$$

Denne typen interpolasjon klarer som regel å bevare flere detaljer enn bilinear interpolasjon, men som man kan se er den også en del mer regnekrevende. I tillegg kan denne interpolasjonen gi opphav til "glorier" langs kanter.

Vi har nå dannet oss et verktøy for å skalere ned bilder, men før vi går videre til bildepyramider tar vi for oss et fenomen kalt aliasing. Når man skalerer ned bilder med høye frekvenser i gråtoneverdiene risikerer man at vi får andre frekvenser i resultatbildet enn de som finnes i originalen, og det er dette fenomenet som kalles aliasing. Et ekstremt tilfelle av aliasing kan forekomme når man til å begynne med har et bilde med mange linjer liggende tett inntil hverandre. Ved nedskalering kan man risikere at linjene ikke blir tatt vare på riktig, og man kan ende opp med merkelige mønstre. Et annet, og kanskje litt mer vanlig eksempel er de hakkete kantene man får, hvor det før skalering var en rett kant. For å unngå slike problemer kan man benytte seg av såkalt antialiasing før skalering. I figur 2.8a kan man se



(a) Denne figuren viser eksempel på aliasing ved nedskalering og bruk av et lavpassfilter for anti-aliasing. Her er bildene nedskalert til halv størrelse langs både x- og y-akse. (b) Viser et lavpassfilter.

Figur 2.8

effekten av nedskalering med og uten bruk av anti-aliasing. Anti-aliasing går ut på å benytte et lavpassfilter på bildet før skalering. Resultatet etter bruk av et lavpassfilter blir et mer diffust bilde, noe som innebærer at de høye frekvensene blir redusert eller helt fjernet. Et eksempel på et lavpassfilter er vist i figur 2.8b. Ved å benytte dette filteret på et bilde medfører det at en piksel $p_a(x, y)$ i det nye bildet, blir funnet ved

$$p_a(x, y) = \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 p(x+i, y+j)$$

hvor $p(x, y)$ er en piksel i det gamle bildet.

2.3.2 Bildepyramider

En bildepyramide [23] er en samling av bilder med ulike oppløsninger. Her er disse bildene arrangert slik at det største bildet ligger i bunn av pyramiden, og når man beveger seg oppover i pyramiden blir oppløsningen til bildene mindre og mindre. Se figur 2.9.

Det finnes to ulike typer bildepyramider. Den første typen, som kalles en tilnærmingspyramide, består av bilder hvor et bilde på nivå j er nedskalert med en gitt faktor, for eksempel 2 langs både x- og y-aksen, fra bildet på nivå $j+1$. Den andre typen kalles en resterende-prediksjonspyramide. Dersom man har et bilde på nivå j og $j+1$ i en tilnærmingspyramide, og kaller disse henholdsvis T_j og T_{j+1} , finner man nivå $J+1$ i en resterende-prediksjonspyramide ved å først oppskalere T_j med en gitt faktor. Deretter lar man bildet på nivå $j+1$ i den resterende-prediksjonspyramiden bli differansen mellom T_{j+1} og oppskaleringen av T_j .

I denne oppgaven fokuserer vi på tilnærmingspyramider, og vi tar nå for oss en nærmere gjennomgang av fremgangsmåten for å lage disse. Vi starter med et bilde, og legger dette i bunnen av pyramiden, som vi kaller nivå J . I [23] legger Gonzales og Woods opp til at dette bildet er $N \times N$ stort, slik at grunnflaten i pyramiden er kvadratisk og bildet i det øverste nivået blir

1×1 stort. Dette er imidlertid ikke et krav, og man kan like gjerne starte med et bilde av størrelsesorden $M \times N$ hvor $M \neq N$. I dette tilfellet ender man dermed opp med et øverste nivå hvor bildestørrelsen er $1 \times \lfloor N/2^J \rfloor$ hvis $N > M$, og $\lfloor M/2^J \rfloor \times 1$ i det motsatte tilfellet. Strukturen er ikke lenger en "perfekt pyramide", i geometrisk forstand, men utgjør en like god representasjon av et bilde med flere oppløsninger.

Hvis man antar at man har et bilde liggende på det nederste nivået, J , i pyramiden med størrelse $M_J \times N_J$, finner man bildet på nivå $J - 1$ ved å nedskalere bildet på nivå J med en gitt faktor v . Det betyr at bildet på nivå $J - 1$ får størrelse $M_{J-1} \times N_{J-1}$, hvor $M_{J-1} = M_J/v$ og $N_{J-1} = N_J/v$. Ved å videreføre dette finner vi bildet på nivå j ved å skalere ned bildet på nivå $j + 1$ med en gitt faktor v . Størrelsen på bildet i nivå J blir dermed $M_{j+1}/v \times N_{j+1}/v$. I figur 2.9 finner man et eksempel på en pyramide hvor originalbildet er $N \times N$ stort.

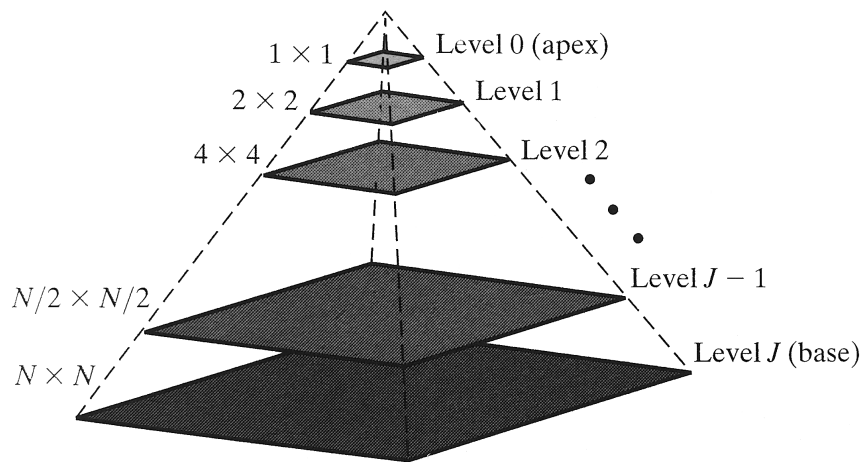
Som man kan se bestemmes antall nivåer i pyramiden etter størrelsen på originalbildet, og man kan finne dette antallet ved

$$J = \lfloor \log_2(\min(N, M)) \rfloor + 1$$

Her antar vi at hvis man nedskaleres et bilde som er $M_j \times N_j$ stort, med en faktor på 2, blir det nedskalerte bildet $\lfloor M_j/2 \rfloor \times \lfloor N_j/2 \rfloor$ stort.

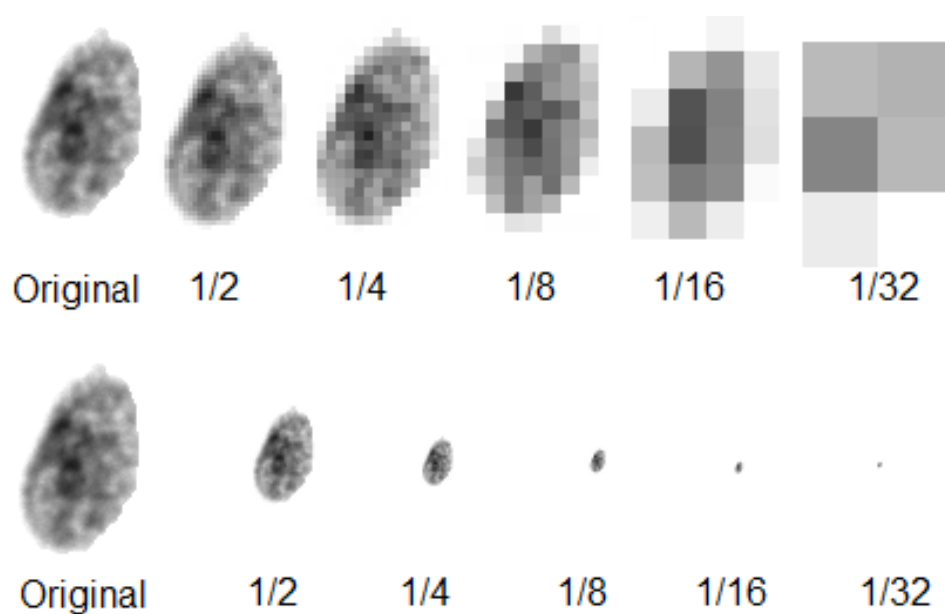
Det er forskjellige måter å filtrere bildene på før nedskalering, slik at man oppnår forskjellige tilnærminger til de nedskalerte bildene. Ved å bruke ulike typer filtrering oppnår man ulike typer bildepyramider. De tre vanligste bildepyramidene er subsampling-pyramide, Gauss-pyramide og middelverdi-pyramide. En subsampling-pyramide får man når det ikke brukes noe filter før nedskalering. Bruker man derimot et gaussisk lavpassfilter eller et middelverdifilter på bildet før nedskalering får man henholdsvis en Gauss- eller middelverdi-pyramide.

En siste ting det er verdt å merke seg om bildepyramider er at man i de fleste tilfeller ikke trenger å ta med de øverste nivåene i pyramiden. Nedskaleres man et bilde flere ganger vil en piksel inneholde informasjon fra et større og større nabolag, og man vil miste informasjon fra flere og flere piksler. Etter gjentatte nedskaleringer vil man komme til et punkt hvor strukturell informasjon blir tilnærmet borte i forhold til originalbildet. Hvis man for eksempel har et bilde med størrelse 512×512 , og man danner en tilnærmingspyramide med maksimalt antall nivåer vil det øverste nivået i pyramiden bestå av et bilde med 1 piksel, og vil være totalt uten strukturell informasjon. Av den grunn lar man det øverste nivået i pyramiden bestå av det minste bildet man kan tillate både i forhold til informasjonen man trenger fra det og problemet man tar for seg. Dersom man for eksempel ser på cellekjerne-bilder, og man ønsker å analysere teksten i cellekjernen, er det ikke noe poeng i å ta med nivåer med bilder hvor kun noen få piksler utgjør selve cellekjernen. Figur 2.10 illustrerer 6 nivåer av et cellekjerne-bilde, og



Figur 2.9: Denne figuren viser en bildepyramide hvor bildet på det nederste nivået er $N \times N$ stort. Bildet er hentet fra [23].

som man kan se, er det lite informasjon å hente ut fra de minste bildene, mens det ser ut til at første nedskalerte bilde er interessant.



Figur 2.10: Denne figuren viser et cellekjerne-bilde sammen med 5 nedskaleringer av dette. Hvert nedskalerte bilde er funnet ved å skalere ned bildet til venstre med en skaleringsfaktor lik 2. Under hvert nedskalerte bilde ser man hvor liten hver akse er i det nedskalerte bilde i forhold til originalbildet. Den minste cellekjernen består av 5 piksler. Den øverste bildeserien viser de nedskalerte bildene med lik størrelse, og bildeserien under viser cellekjernene i faktisk størrelse.

2.4 Klassifikatorer

Når man skal klassifisere data, er man nødt til å ha en klassifikator, og i følge Jain et al. [31] er de fire mest kjente fremgangsmåtene sammenlikning av maler, statistisk klassifikasjon, syntaktisk eller strukturell sammenlikning og nevralt nettverk. Ettersom vi anvender statistiske egenskaper i denne oppgaven velger vi å begrense oss til statistisk klassifikasjon.

Ved statistisk klassifikasjon er det vanlig at hver klasse blir representert med en n -dimensjonal egenskapsvektor. Det er viktig å velge egenskaper som skiller klassene godt, slik at man unngår for stor n og "curse of dimensionality" [29]. "Curse of dimensionality" er et fenomen som går ut på at en klassifikator blir bedre jo flere egenskaper man bruker, inntil man når et kritisk antall egenskaper. Når man når det kritiske antallet faller nemlig klassifiseringsraten til klassifikatoren. Dette kommer av at klassifikatoren blir overtilpasset på trening, altså blir skillelinjen mellom klassene for spesialisert i forhold til datapunktene man har til trening, og man klarer ikke å overføre dette til det generelle tilfellet. Se figur 2.11 for en illustrasjon av fenomenet. Det man ønsker å oppnå med statistisk klassifikasjon er å finne en bestemmelseslinje som skiller klassene best mulig.

2.4.1 Minimum distanse klassifisering

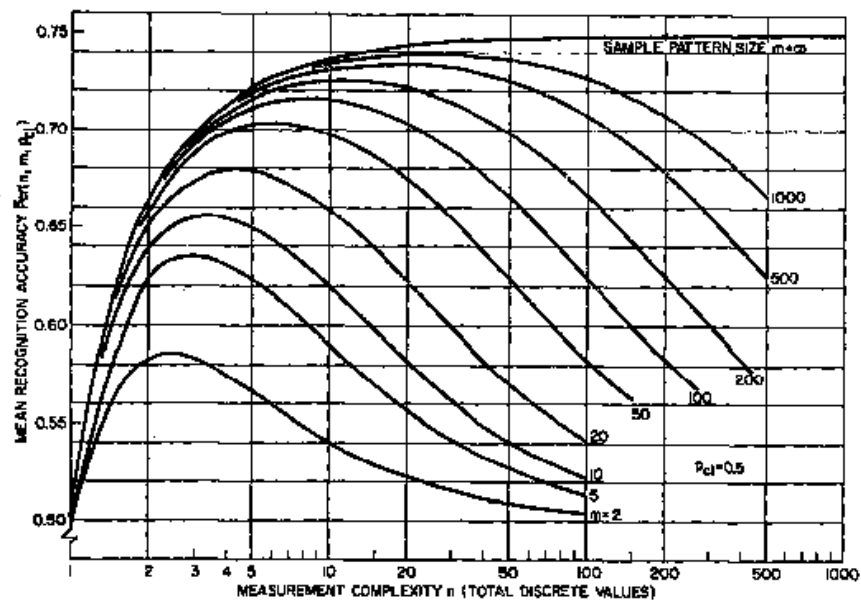
En veldig enkel klassifikator er minimum distanse klassifisering [23]. Her starter man med å beregne forventningsverdien til de klassene man arbeider med. Når man så tar for seg et nytt datapunkt d , finner man avstanden mellom dette datapunktet og forventningsverdiene til de ulike klassene. Til slutt blir datapunktet klassifisert til klasse ω_i hvis og bare hvis

$$|\mu_i - p| \leq |\mu_j - p|, \quad \forall \mu_j, \quad i \neq j$$

hvor μ_i er forventningsverdien til klasse ω_i . Denne metoden vil fungere godt når det er stor avstand mellom forventningsverdiene, og når spredningene ikke overlapper for mye. I tillegg er den lite regnekrevende. Metoden tar imidlertid ikke hensyn til varians, og vil derfor anta at alle klassene varierer like mye. Hvis man står overfor klasser med ulik varians, og de samtidig er forholdsvis nærme hverandre, kan det være en god ide å bruke en annen klassifikator som tar hensyn til dette.

2.4.2 Nærmeste nabo klassifisering

Nærmeste nabo klassifisering [12] er, i likhet med minimum distanse klassifisering, en ganske enkel og rett frem metode for klassifisering. Her regner man imidlertid ikke ut noen forventningsverdi. I stedet sier man at et nytt datapunkt d får samme klasse som det nærmeste punktet i treningsettet. Når man skal finne det nærmeste punktet er man nødt til å beregne avstanden til alle punktene i treningsettet. Det er vanlig å bruke det



Figur 2.11: Illustrasjon på "curse of dimensionality" [29]. Midlere korrekt klassifikasjonsrate (vertikal akse) er vist som funksjon av antall egenskaper (horisontal akse) og antall datapunkter (gitt på hver kurve) for et to-klasse problem der apriori-sannsynlighetene er like. Dersom vi ser på kurven for $m = 100$ datapunkter, ser vi at klassifiseringsraten øker frem til vi har 7-10 egenskaper hvor klassifiseringsraten er høyest. Når antall egenskaper blir flere enn 10 synker klassifiseringsraten. Figuren er hentet fra [29].

Euklidske distanse målet, se seksjon 2.2.5, for å beregne avstandene. Denne metoden kan også utvides slik at man ser på de k nærmeste naboene. Det som kan være vanskelig med denne metoden er å bestemme hva k skal være. Setter man k til å være for liten, blir klassifikatoren veldig sensitiv til støy. Dersom man på den andre siden setter k til å være for stor, risikerer man å se på datapunkter som er veldig langt unna det datapunktet som skal klassifiseres, hvilket medfører større sannsynlighet for feilklassifisering.

2.4.3 Bayesisk Klassifisering

Ved å bruke Bayes regel [17] kan man klassifisere et datapunkt til den klassen det er mest sannsynlig at datapunktet tilhører [46]. Sannsynligheten beregnes på grunnlag av en n -dimensjonal egenskapsvektor \mathbf{x} . Bayes regel er gitt ved

$$P(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)P(\omega_i)}{\sum_{j=1}^k p(\mathbf{x}|\omega_j)P(\omega_j)}$$

Hvor ω_j er klasse j og \mathbf{x} er den n -dimensjonale egenskapsvektoren. Ved klassifiseringen bestemmer man at datapunktet tilhører ω_i når

$$P(\omega_i|\mathbf{x}) > P(\omega_j|\mathbf{x}) \quad \forall \quad j \neq i$$

I følge Gonzlez og Woods [23] er det mest vanlig å anta normalfordeling for $p(\mathbf{x}|\omega_i)$. Dette følger ganske naturlig siden normalfordelingen er en god tilnærming til mange naturlige fenomen [24].

Multivariabel Gaussisk Klassifikator

Når man antar normalfordeling for $p(\mathbf{x}|\omega_i)$ i Bayesisk klassifisering, og man har $k > 1$ klasser, blir klassifikatoren kalt multivariabel Gaussisk klassifikator. I det k -dimensjonale tilfellet er $p(\mathbf{x}|\omega_i)$ gitt ved [17]

$$p(\mathbf{x}|\omega_i) = \frac{1}{(2\pi)^{\frac{k}{2}} |\Sigma_i|^{\frac{1}{2}}} e^{-1/2(\mathbf{x}-\mu_i)^T \Sigma_i^{-1} (\mathbf{x}-\mu_i)}$$

Her er μ_i en vektor med forventningsverdiene til egenskapene til klasse ω_i og Σ_i er den tilhørende kovariansmatrisen.

2.4.4 Støttevektormaskin (SVM)

Støttevektormaskin, "Support Vector Machine", SVM [11] er en klassifikator for hovedsakelig 2 klasser. I tillegg til å skille de to klassene lineært, prøver SVM å maksimere avstanden fra klassene til beslutningsgrensen og på den måten minimere klassifiseringsfeilen. Dette er igjen med på å gjøre klassifikatoren mer robust. I tilfeller hvor klassene ikke er lineært separable kan man innføre en kostnadsfunksjon, som innebærer at man innfører en kostnad for hver feilklassifisering. Deretter prøver man å minimere denne kostnadsfunksjonen, slik at man oppnår en best mulig klassifisering. Man kan også benytte seg av det såkalte kjernetrikset [40], som går ut på at man benytter en transformasjon, slik at dataen kan samle seg bedre. Dette kan

også medføre at det blir større avstand mellom klassene, noe som bedrer klassifikasjonen. Til slutt er det verdt å tilføye at selv om SVM hovedsakelig er ment for 2 klasser, er det også mulig å utføre klassifisering for flere klasser [40]. Dette kan blant annet gjøres ved å utføre såkalte alle-mot-en klassifiseringer, som innebærer at man plukker ut en klasse, og sier at alle de andre klassene er en annen klasse. Deretter gjør man dette for alle klassene, for på den måten å finne de beste skillene mellom dem.

2.4.5 Oppsummering

Vi har nå sett på ulike klassifikatorer, og de ulike klassifikatorene blir brukt til forskjellige typer problemer. Dersom det er stor avstand mellom forventningsverdiene til hver klasse, og klassene har relativt lik varians, vil minimum distanse klassifisering kunne bli brukt som en klassifikator. Det er også verdt å tenke på at dersom man skal velge denne klassifikatoren, bør spredningene til hver klasse ikke overlappe for mye. Vi skrev også om nærmeste nabo klassifisering, og det kom frem at det kan være vanskelig å bestemme hvor mange naboer man skal se på. Siden metoden lett kan bli et offer for "curse of dimensionality" velger vi å se bort fra denne klassifiseringsmetoden videre i oppgaven. Ved multivariabel gaussisk klassifisering beregner man hvilken klasse det er mest sannsynlig at et datapunkt hører til. Denne sannsynlighetsfordelingen baseres på forventningsverdi og kovarians, og er et godt alternativ når man står overfor klasser med ulik varians i egenskapene. Ettersom normalfordelingen er en god tilnærming til mange naturlige fenomener, kan det se ut til at denne klassifikatoren kan være et godt valg for oss. SVM var den siste klassifikatoren vi så på. Denne metoden er hovedsakelig beregnet for 2 klasser, og på den måten passer den godt for vårt problem. Dersom klassene er relativt godt separert, vil det være greit å kunne bruke SVM. Skulle derimot klassene overlappe mye må man benytte kjernetrikset, og det kan være vanskelig å velge hvilken kjernefunksjon som blir optimal å bruke.

Kapittel 3

Prosjektet - datasettet

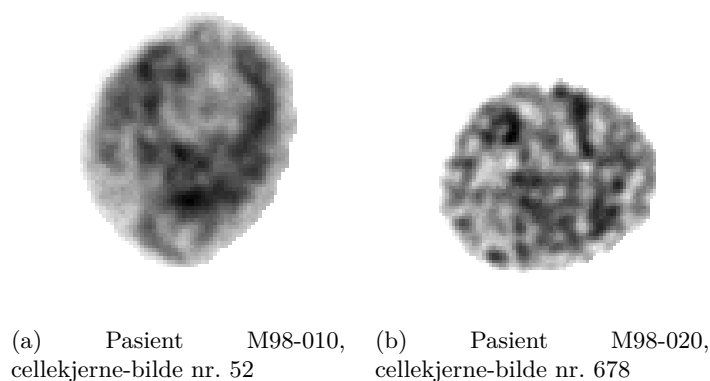
3.1 Datasettene

Vi benytter oss av en del av et datasett som består av bilder av cellekjerne i kreftsvulster i prostata, kalt M98 og et datasett bestående av cellekjernebilder fra kreftsvulster i eggstokkene, kalt L23. I begge disse datasettene har pasientene blitt fulgt opp i minst ti år. Begge datasettene er basert på monolayer-bilder, hvilket vil si at hele cellekjernen er presset litt sammen og deretter blitt avbildet med et mikroskop.

3.1.1 Prostata-materialet

Prostata-materialet (M98) er del av en fortløpende serie pasienter operert (radikal prostatektomi) for prostatakreft ved Radiumhospitalet i perioden 1984-2005. Datasettet består av totalt 316 pasienter og er delt opp i et trening- og testsett ved å velge annenhver pasient til de to datasettene. Pasientene med god prognose har blitt fulgt opp i ti år uten tilbakefall, mens pasientene med dårlig prognose har fått tilbakefall innen seks år. I denne oppgaven konsentrerer vi oss imidlertid av to såkalte supertreningsett, hvor vi lar det ene bli brukt til trening, mens det andre blir forbeholdt test. Supertreningsettene består av pasienter med veldig god og veldig dårlig prognose, og utgjør ytterpunktene i det originale M98. For at en pasient skal havne i et av supertreningsettene, er det nødt til å ikke være noe rest av tumoren etter operasjon. Supertreningsettene består av 34 pasienter, hvor treningsettet består av 10 pasienter med god prognose og 10 med dårlig prognose, og testsettet består av 8 pasienter med god prognose og 6 pasienter med dårlig prognose. Tanken bak supertreningsettene er å kunne finne typiske teksturmønstre som er ulike i de to prognosegruppene. Når vi i denne oppgaven skriver om M98, er det supertreningsettene vi refererer til, med mindre noe annet er spesifisert.

Prostata blir lagt i formalin og delt opp i biter ved makrobeskjæring, for deretter å bli støpt inn i parafinblokker. En patolog vurderer snitt fra hver av blokkene og utfører blant annet en Gleason gradering [22]. En slik gradering er et mål på aggressiviteten, og er en god prognostisk markør for prostatakreft. I M98 blir den blokken med høyest Gleason gradering, altså

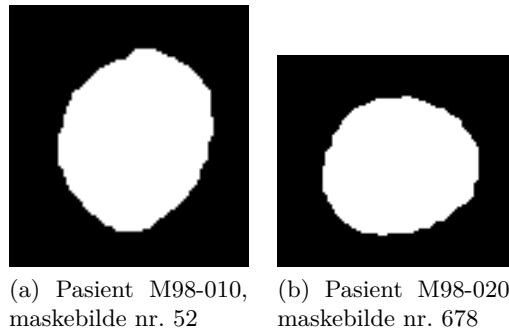


Figur 3.1: Eksempler på to cellekjerne. Pasient M98-010 har dårlig prognose, mens pasient M98-020 har god prognose.

den blokken som er mest aggressiv, valgt ut for å studere teksturen.

For å utarbeide monolayers ble det brukt en modifikasjon av Hedleys metode [14]. Monolayerne består av isolerte cellekjerne fra en eller flere snitt av tykkelse $50\mu\text{m}$. Ved å følge en etablert protokoll [62], ble cellekjernene farget etter Feulgen-Schiff metoden. For å avbilde hvert bildeområde ble Ploidy Work Station (Room 4, East Sussex, UK) brukt. Dette systemet består av et Zeiss Imager A2 mikroskop utstyrt med en 40/0.75 objektivlinse (Zeiss), et 546 nm grønt filter og et svart/hvitt høyoppløselig digitalkamera (Zeiss AxioCam MRm Rev.3) med 1388×1040 piksler og en gråtoneoppløsning på 10 bit/piksel.

Det er verdt å påpeke at selv om pasienter i gruppen med dårlig prognose vil ha cellekjerne som oppfører seg deretter, vil flertallet av cellekjernene kunne oppføre seg som cellekjernene til en pasient med god prognose [15]. Av den grunn er man nødt til å ha tilstrekkelig mange cellekjerne-bilder, og man kan forvente vanskeligheter med å klassifisere flere av pasientene riktig. Hver pasient har en mappe med et gitt antall cellekjerne-bilder og hvert av disse bildene har størrelser rundt 95×95 piksler, hvor oppløsningen er 162 nm per piksel. Selve cellekjernen opptar ca 33 % av bildet, hvilket medfører at vi analyserer ca. 3000 piksler per bilde. Et par slike bilder er illustrert i figur 3.1. Til alle disse cellekjerne-bildene har vi også tilhørende maskebilder. Maskebildene er binære bilder, hvor det hvite området i bildet viser hvor cellekjernen befinner seg, og det er kun dette området som er av interesse i våre beregninger. Eksempel på maskebilder finnes i figur 3.2.

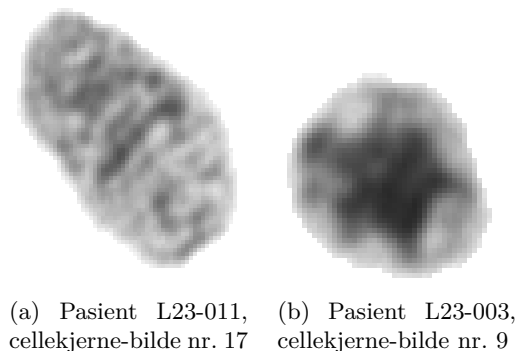


Figur 3.2: Eksempler på to maskebilder. Pasient M98-010 har dårlig prognose, mens pasient M98-020 har god prognose.

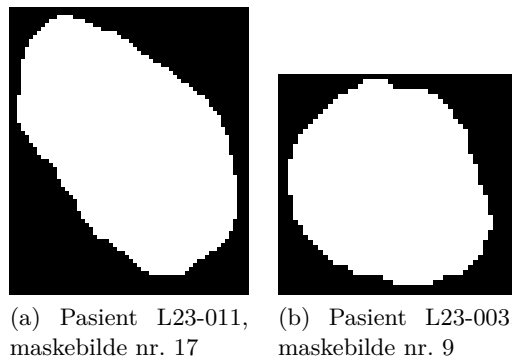
3.1.2 Eggstokk-materialet

Eggstokk-materialet (L23) består av vevsprøver hentet fra 134 pasienter som hadde eggstokkreft klassifisert som FIGO stadium 1 [50]. Pasientene som tilhører gruppen for god prognose, er de som overlevde oppfølgingsperioden uten tilbakefall. De pasientene som havnet i gruppen for dårlig prognose fikk enten tilbakefall eller døde av en kreftrelatert sykdom. Vevsprøvene har blitt fiksert i 4 % bufret formalin og innstøpt i parafin. Tumorvevet har blitt valgt ut av en patolog. Ved å bruke en modifikasjon av Hedleys metode [14] ble det utarbeidet monolayers bestående av isolerte cellekjerne fra en eller flere snitt med tykkelse $50\mu\text{m}$. Ved å følge en etablert protokoll [62], ble cellekjernene farget etter Feulgen-Schiff metoden.

For å avbilde hvert bildeområde ble Fairfield DNA Ploidy systemet (Fairfield Imaging Ltd, Nottingham, UK) brukt. Dette systemet består av et Zeiss Axioplan mikroskop utstyrt med en 40/0.75 objektivlinse (Zeiss), et 546 nm grønt filter og et svart/hvitt høyoppløselig digitalkamera (C4742-95, Hamamtsu Photonics K.K., Hamamtsu, Japan) med 1024×1024 piksler og en gråtoneoppløsning på 10 bit/piksel. For hvert bildefelt ble det utført skyggekorrigering. Hver cellekerne ble segmentert i forhold til en global terskel. I tillegg ble cellekerne-bildene gått gjennom manuelt, slik at man luket bort



Figur 3.3: Eksempler på to cellekjerne. Pasient L23-011 har dårlig prognose, mens pasient L23-003 har god prognose.



Figur 3.4: Eksempler på to maskebilder. Pasient L23-011 har dårlig prognose, mens pasient L23-003 har god prognose.

stromaceller, nekrotiske og ufullkomne cellekjerne. Gjennomsnittlig har vi 281 cellekjerne per pasient. Det minste antallet cellekjerne for en pasient er 220, mens det høyeste antallet cellekjerne for en pasient er 314. Selve cellekjernen utgjør i gjennomsnitt 59 % av cellekjerne-bildene, som har en gjennomsnittlig størrelse på 80×80 piksler med en oppløsning på 166 nm per piksel. Dette gjør at en cellekjerne i L23 består av ca 3800 piksler.

L23 er delt opp i et trening- og testsett. Hvert sett består av 67 pasienter. Treningsettet består av 49 pasienter med god prognose og 18 pasienter med dårlig prognose, mens testsettet består av 48 pasienter med god prognose og 19 med dårlig prognose. Det ble tilfeldig avgjort hvilke pasienter som skulle tilhøre hvert av de to datasettene, ved velge annen hver pasient. Se figur 3.3 for eksempler på cellekjerne fra L23, og figur 3.4 for tilhørende maskebilder. Maskebildene blir brukt på lik måte som for M98.

3.2 Tidligere resultater

Jørgensen et al. [32] oppnådde en korrekt klassifiseringsrate (CCR) på 87 %. CCR er et vektet gjennomsnitt av riktig andel klassifisert god prognose og riktig andel klassifisert dårlig prognose, hvor vektingen er andelen av hver gruppe. I denne studien ble det benyttet GLCM, GLRLM og GLEM, og hele 275 egenskaper og en sample/egenskap rate på 2.5. Mairinger et al. [38] brukte GLCM, GLRLM og fraktal-parametre. De brukte 114 egenskaper og en sample/egenskap rate på 8.3, og oppnådde en CCR på 92 %. I [42] oppnådde Mohamed et al. en CCR på 89 %. Det ble i denne studien benyttet GLCM, og 65 egenskaper og en sample/egenskap rate på 4.8. Tidligere studier av statistisk teksturanalyse benyttet på prostata-materiale har vist ganske gode resultater. I alle de nevnte studiene har det blitt benyttet en klassifikator som er basert på et treningsett, som deretter er blitt validert på et eget testsett.

Når man går over til å se på tidligere oppnådde resultater av statistisk teksturanalyse benyttet på eggstokk-materiale, ser man at resultatene er

dårligere enn for prostata-materialet. Nielsen et al. [47] oppnådde CCR på 63 % ved å benytte egenskaper basert på fraktaler. Det viste seg å være stor overlapp mellom god og dårlig prognose, og de konkluderte med at egenskapene ikke var gode nok for klassifisering. Bedre resultater oppnådde Nielsen et al. i [50]. Det beste resultatet i denne studien ble oppnådd med en adaptiv GLCM egenskap, og ga en CCR på 70 %. Dette er et godt resultat, men det er fortsatt ønskelig å oppnå bedre resultater slik at metodene kan benyttes klinisk. Nielsen et al. oppnådde også en CCR på rundt 70 % i [48], men denne gangen med adaptive egenskaper fra entropimatrise-metoden GLEM.

Kapittel 4

Design av eksperimentet

I dette kapitlet tar vi for oss fremgangsmåter som omhandler bruken av sannsynlighetsfordelinger basert på TU-forekomster og adaptive egenskaper for pasienter og prognosegrupper. Etter at slike sannsynlighetsfordelinger er satt opp finner vi et mål på forskjellen mellom de ulike sannsynlighetsfordelingene. Tanken bak dette er å forhåpentligvis finne små forskjeller mellom pasientene i lik prognosegruppe, og større forskjeller mellom pasienter i de to ulike prognosegruppene.

Kapitlet starter med tre ulike anvendelser av LBP. To av dem anvender henholdsvis 8 og 16 piksels nabolag på originale cellekjerne-bilder, mens den siste går ut på å bruke 8 piksels nabolag på nedskalerte bilder. Hver av dem har blitt tilegnet en egen seksjon. I hver av disse seksjonene vil man finne tilhørende bruk av varians, samt kvantisering av TU, oppsett av sannsynlighetsfordelinger og måling av ulikhet mellom sannsynlighetsfordelingene. Operatorene benyttet på 8 piksels nabolag i original-bildene setter standarden for valg av distansemåler og klassifikator. Dette gjøres for at resultatene skal ha likt sammenlikningsgrunnlag. Avslutningsvis kommer det en seksjon hvor vi går inn på anvendelsen av adaptive metoder.

4.1 LBP benyttet på nabolag bestående av 8 piksler i originale cellekjerne-bilder

I denne seksjonen gjennomgår vi $LBP_{8,1}$. Vi husker fra seksjon 2.1.7 at den generelle formelen for LBP er gitt ved

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p$$

hvor P er antall naboer, R avstanden fra senterpikselet g_c til nabopikselene g_p og s er gitt ved

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

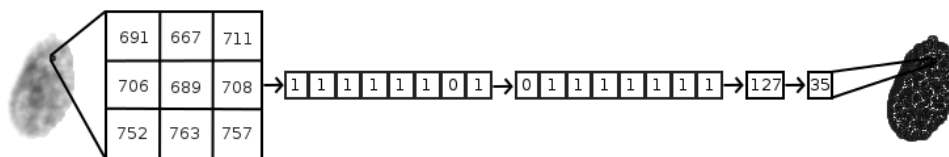
Tabell 4.1: Mappingen av de 36 unike rotasjonsinvariante TU-mønstrene vist i figur 2.3

TU	0	1	3	5	7	9	11	13	15	17	19	21	...
(*)	1	2	3	4	5	6	7	8	9	10	11	12	...
23	25	27	29	31	37	39	43	45	47	51	53	...	
13	14	15	16	17	18	19	20	21	22	23	24	...	
55	59	61	63	85	87	91	95	111	119	127	255		
25	26	27	28	29	30	31	32	33	34	35	36		

For å analysere et 8 piksels nabolag vil den naturlige LBP-operatoren være $LBP_{8,1}$, selv om vi i teorien også kunne skrevet om bruk av et 8 piksels nabolag lenger fra senterpikselen, for eksempel ved bruk av $LBP_{8,1.5}$ eller $LBP_{8,2}$. Til tross for mange ulike muligheter holder vi oss til $LBP_{8,1}$. Dette er fordi pikslene i et slikt nabolag ligger så tett som mulig, og gir dermed et mer samlet bilde av den lokale teksten. I tillegg er dette en operator som er godt testet av blant annet Ojala et al. [54] [56] [53] og Ma [36].

Som indikert ser $LBP_{8,1}$ på de 8 nærmeste pikslene til et senter-piksel, og gir ut en sekvens bestående av 8 binærtall. Denne sekvensen gjøres så om til et heltall i intervallet $[0 \ 255]$, og denne verdien kaller vi "Texture Unit" (TU). Vårt formål med en TU er å se på sannsynlighetsfordelingen basert på andelen forekomster av denne i cellekjernene, for deretter å måle avstanden mellom pasienter og pasientgrupper. Ettersom dette skal gjøres for hver av de 256 TU er vi nødt til å redusere dette antallet. Vi starter med å gjøre LBP-operatoren rotasjonsinvariant. Som beskrevet i seksjon 2.1.7 oppnår vi rotasjonsinvarians ved å rotere den binære sekvensen slik at den resulterende sekvensen har lavest verdi i forhold til alle sekvensene det er mulig å oppnå ved rotasjon. På denne måten reduseres antall forskjellige utverdier til 36. Vi utfører en 1-til-1 mapping av disse verdiene inn i intervallet $[1, 36]$, hvor TU-verdiene har blitt rangert i stigende rekkefølge, se tabell 4.1. I denne tabellen representerer (*) de nye verdiene etter mappingen.

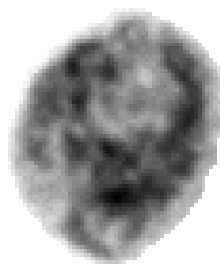
Denne LBP-operatoren er blitt benyttet på alle cellekjerne-bilder, og for hvert cellekjerne-bilde er det laget et nytt tilhørende bilde bestående av TU-verdiene funnet i cellekjerne-bildene. Et punkt (i, j) i det nye bildet I_{TU} blir funnet ved å benytte LBP-operatoren på cellekjerne-bildet, I_c , ved å la senterpikselen være $I_c(i, j)$. Mens originalbildene har 1024 gråtoner, blir TU-bildene lagret med maksimalt 36 ulike verdier. Figur 4.1 illustrerer hvordan TU-bilder blir laget fra et cellekjerne-bilde. Dersom senter-pikselen er i den ytterste kanten av cellekjernen, slik at noen av nabopikslene ikke tilhører cellekjernen, velger vi å se bort fra denne pikselen i beregningen. På denne måten unngår vi å introdusere kunstig lagde elementer. Dette medfører imidlertid at vi gjennomsnittlig mister 8.5 % av informasjonen i cellekjernen i M98 og 8.1 % i L23. Eksempler på TU-bilder er å se i figur 4.2 og 4.3.



Figur 4.1: Når man skal generere et TU-bilde starter man med et cellekjerne-bilde lik det til venstre. $LBP_{8,1}$ benyttes på hver piksel i dette bildet. Til høyre for cellekjerne-bildet kan man se gråtoneverdiene til det lokalt nabolaget $LBP_{8,1}$ benyttes på. Dette resulterer i den binære sekvensen som er å finne til høyre for det lokale nabolaget. Når man har en rotasjonsinvariant LBP-operator, roterer man bit-sekvensen til man oppnår sekvensen med lavest magnitude. I dette tilfellet gjør denne rotasjonen at vi ender opp med bit-sekvensen lengst til høyre. Denne bit-sekvensen oversettes så til et tall i titallsystemet, som i dette tilfellet er 127. Etter 1-til-1 mappingen ender man opp med 35, som igjen settes inn på aktuell plass i TU-bildet.

4.1.1 Fordelingen av de ulike TU-numrene

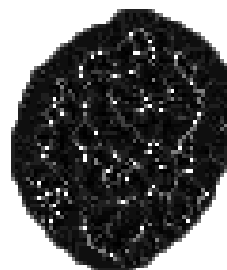
Som Ojala et al. [54] er inne på forekommer noen TU-numre langt oftere enn andre. For å få et inntrykk av hvordan de 36 forskjellige TU-numrene fordeler seg, setter vi opp et normalisert histogram over hvor ofte hver av de forekommer totalt over alle pasienter i en gruppe. Figur 4.4 viser slike histogrammer, for god og dårlig prognose. Alle dataene er hentet fra datasettet M98. I figur 4.5 kan man se tilsvarende for L23. Det første man kan legge merke til er den store likheten mellom klassene, noe som vil gjøre klassifiseringen vanskelig, og det andre er at noen TU-numre forekommer langt oftere enn andre. Dette er blant annet TU-nummer 1, 2, 3, 5, 9, 17, 28, 35 og 36, som er de 9 TU-numrene med de tilhørende binære mønstrene som er mest homogene. Når vi skriver om homogeniteten til de binære mønstrene, tenker vi på antall 0/1 overganger som forekommer i en sekvens med binære tall. Dette er i samsvar med Ojala et al. [54] sine erfaringer, som observerte at ca. 90 % av alle TU-numre vil være en av disse 9, når man ser på et 8 piksels nabolag. Dersom man adderer sammen de normaliserte sannsynlighetene for de 9 mest homogene mønstrene i figur 4.4a, 4.4b, 4.5a og 4.5b ender vi også opp med at 90 % av TU-numrene er en av de 9 mest homogene. I figur 4.4c og 4.5 kan man se at differansene finnes i de 9 mest homogene mønstrene (1, 2, 3, 5, 9, 17, 28, 35, 36), men også i for eksempel TU-nummer 4 og 32. På grunnlag av dette vil det lønne seg å bruke en gruppering av TU-numrene som tar hensyn til de 9 mest homogene, men også de andre mønstrene med differanse (blant annet 4 og 32). Vi observerer imidlertid et par ulikheter mellom differansene i M98 og L23. Differansen mellom TU-gruppe 17 i god og dårlig prognose i L23 er veldig lav sammenliknet med tilsvarende differanse i M98, mens differansen mellom TU-gruppe 5 i de to prognosegruppene er høyere i L23 enn i M98. Vi observerer også at flere av TU-gruppene med lave differanser i L23 har høyere differanser i M98.



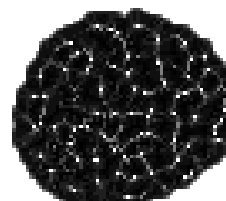
(a) Pasient M98-010, dårlig prognose



(b) Pasient M98-020, god prognose



(c) Pasient M98-010, dårlig prognose

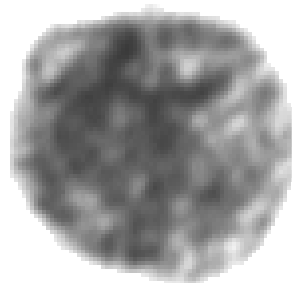


(d) Pasient M98-020, god prognose

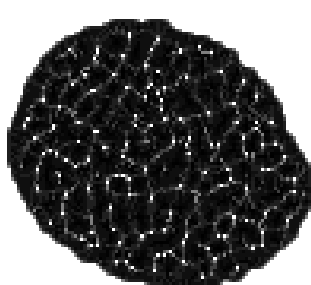
Figur 4.2: 4.2a og 4.2b er eksempler to cellekjerne-bilder fra M98, mens 4.2c og 4.2d er eksempler på tilhørende TU-bilder fra M98.



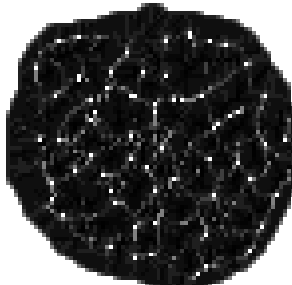
(a) Pasient L23-027, dårlig prognose



(b) Pasient L23-082, god prognose

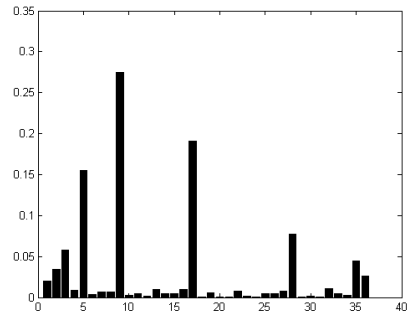


(c) Pasient L23-027, dårlig prognose

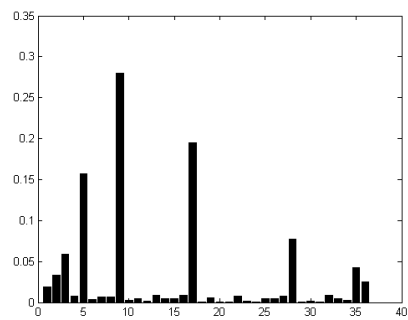


(d) Pasient L23-082, god prognose

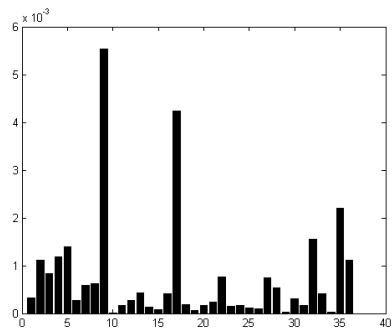
Figur 4.3: 4.3a og 4.3b er eksempler på to cellekjerne-bilder fra L23, mens 4.3c og 4.3d er eksempler på tilhørende TU-bilder fra L23.



(a) God prognose

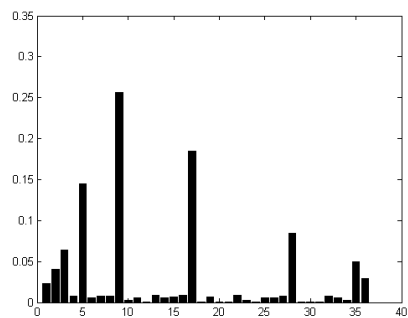


(b) Dårlig prognose

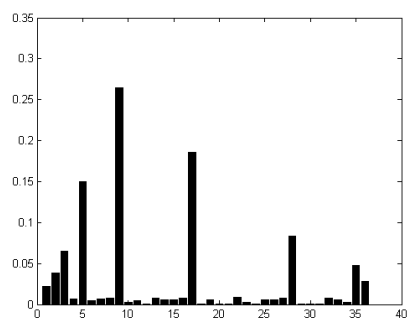


(c) Differansen mellom prognosegruppene

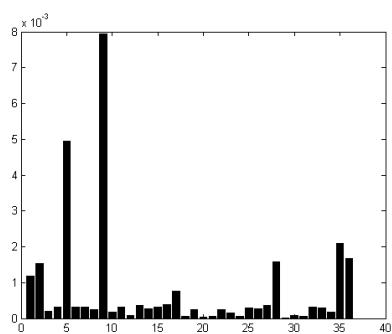
Figur 4.4: Normaliserte histogram over fordelingen av TU-numrene for alle celler i hele M98. Histogrammene 4.4a og 4.4b baseres på henholdsvis pasienter med god og dårlig prognose. 4.4c viser differansen mellom de to prognosegruppene.



(a) God prognose



(b) Dårlig prognose



(c) Differansen mellom prognosegruppene

Figur 4.5: Normaliserte histogram over fordelingen av TU-numrene for alle celler i treningsettet til L23. Histogrammene 4.4a og 4.4b baseres på henholdsvis pasienter med god og dårlig prognose. 4.4c viser differansen mellom de to prognosegruppene.

4.1.2 Reduksjon av antall TU-numre

Som vi var inne på i forrige seksjon er fordelingen av andelen TU-numre skjev. Ojala et al. [54] foreslår å bruke 10 grupperinger av TU-numrene, en gruppe for hver av de 9 TU-numrene med tilhørende binære mønstre som er mest homogene, og en siste gruppe for alle de andre. Dette vil imidlertid medføre tap av informasjon som kan ligge i forekomster av mer komplekse LBP. Ma [36] foreslo å benytte en gruppering som ville ta vare på litt mer av informasjonen fra de inhomogene LBP. Grupperingen går ut på å la de 9 uniforme mønstrene bli egne grupper, slik som Ojala et al. Videre får de LBP som består av like mange 0 og 1 forekomster en egen gruppe. De resterende mønstrene blir gruppert ved å telle opp hvor mange forekomster av 0 og 1 det er i mønstret. Dersom det er to forekomster av 1, og resten er 0, vil alle de inhomogene mønstrene hvor dette forekommer havne i en gruppe. Og man vil få en egen gruppe for de mønstrene hvor det er to forekomster av 0 og resten 1. Denne fremgangsmåten resulterer i 14 TU-grupper. Ved å følge denne grupperingen vil man imidlertid også miste en del informasjon fra de inhomogene mønstrene, og av den grunn velger vi å se på en tredje gruppering.

Den tredje grupperingen ble foreslått av Albregtsen [3], og går ut på å gruppere TU-numrene i forhold til løpelengder og antall 0/1 overganger. Fremgangsmåten går ut på at vi beholder de 9 mest homogene LBP som egne grupper. Deretter ser vi på kompleksiteten (antall 0/1-overganger) samt maks løpelengde i et binært mønster, for så å la alle LBP med lik kompleksitet og maks løpelengde havne i samme gruppe. På den måten vil vi ende opp med de 17 grupperingene som er vist i tabell 4.2. Legg merke til at vi lar TU-numrene og TU-gruppene starte på 1 og ikke 0, slik som TU-verdiene gjør. Dette er kun for å gjøre verdiene litt enklere å jobbe med, ettersom indeksering i MATLAB starter på 1.

Vi har tidligere skrevet om muligheten til å gjøre LBP-mønsteret speilingsinvariant, noe som i kombinasjon med rotasjonsinvarians ville gitt oss 30 TU-numre. Bruk av speilinginvarians ville medført at TU-verdiene 11 og 13, 19 og 25, 23 og 29, 43 og 53, 55 og 59 samt 47 og 61 ville blitt ett TU-nummer hver, i stedet for to. Fra 4.2 ser vi imidlertid at disse parene med TU-verdier havner innenfor samme TU-gruppe, og av den grunn vil det være unødvendig å gjøre $LBP_{8,1}$ speilingsinvariant.

4.1.3 Sannsynlighetsfordeling for TU-gruppene

Vi har skrevet om hvordan fordelingen av TU-gruppene er i de to prognosegruppene, men vi har bare sett på gjennomsnittlig andel forekomst av hver TU-gruppe, og ikke sett hvordan gjennomsnittlig andel varierer fra bilde til bilde. Dette er ubrukt informasjon vi kan benytte oss av, og vi starter med å se på TU-bildene til en pasient p_k . For dette bildet setter vi opp andelen forekomster av de 17 TU-gruppene i en vektor. Dette gjøres så for alle TU-

Tabell 4.2: 17 TU grupperinger.

TU-gr.	TU-verdi	TU-nr	Binært mønster	Antall overganger	Løpe-lengder	Maksimal løpelengde	Gj.snitt løpelengde
1	0	1	00000000	0	8	8	8.0
2	255	36	11111111	0	8	8	8.0
3	1	2	00000001	2	7 1	7	4.0
4	127	35	01111111	2	7 1	7	4.0
5	3	3	00000011	2	6 2	6	4.0
6	63	28	00111111	2	6 2	6	4.0
7	7	5	00000111	2	5 3	5	4.0
8	31	17	00011111	2	5 3	5	4.0
9	15	9	00001111	2	4 4	4	4.0
10	5	4	00000101	4	5 1 1 1	5	2.0
10	95	32	01011111	4	5 1 1 1	5	2.0
11	9	6	00001001	4	4 2 1 1	4	2.0
11	11	7	00001011	4	4 2 1 1	4	2.0
11	13	8	00001101	4	4 2 1 1	4	2.0
11	47	22	00101111	4	4 2 1 1	4	2.0
11	61	27	00111101	4	4 2 1 1	4	2.0
11	111	33	01101111	4	4 2 1 1	4	2.0
12	17	10	00010001	4	3 3 1 1	3	2.0
12	23	13	00010111	4	3 3 1 1	3	2.0
12	29	16	00011101	4	3 3 1 1	3	2.0
12	119	34	01110111	4	3 3 1 1	3	2.0
13	19	11	00010011	4	3 2 2 1	3	2.0
13	25	14	00011001	4	3 2 2 1	3	2.0
13	27	15	00011011	4	3 2 2 1	3	2.0
13	39	19	00100111	4	3 2 2 1	3	2.0
13	55	25	00110111	4	3 2 2 1	3	2.0
13	59	26	00111001	4	3 2 2 1	3	2.0
14	51	23	00110011	4	2 2 2 2	3	2.0
15	21	12	00010101	6	3 1 1 1 1 1	3	1.3
15	87	30	01010111	6	3 1 1 1 1 1	3	1.3
16	37	20	00100101	6	2 2 1 1 1 1	2	1.3
16	43	24	00101011	6	2 2 1 1 1 1	2	1.3
16	45	21	00101101	6	2 2 1 1 1 1	2	1.3
16	53	31	00110101	6	2 2 1 1 1 1	2	1.3
16	91	18	01011011	6	2 2 1 1 1 1	2	1.3
17	85	29	01010101	8	1 1 1 1 1 1 1 1	1	1.0

bildene til alle pasientene i en prognosegruppe ω_i . Når disse vektorene er funnet setter vi dem inn som rader i en matrise M_i . Dette gjøres for begge prognosegruppene, slik at vi ender opp med to matriser, M_0 og M_1 . Disse matrisene vil dermed være $m_i \times 17$ store, hvor m_i er antall bilder som finnes for alle pasienter i gruppen ω_i .

Videre tar vi for oss en enkelt TU-gruppe $\tau \in [1, 17]$, altså konsentrerer vi oss nå kun om kolonne τ i M_i . Vi lar så

$$a_j = M(j, \tau)$$

På denne måten kan man tenke på a_j som sannsynligheten for at en tilfeldig utvalgt TU-gruppe i bilde j er τ . Deretter lar vi m være den maksimale a_j over alle radene i de to matrisene, og i tabell 4.3 er disse maksverdiene gitt. For å sette opp en sannsynlighetsfordeling, lager vi l intervaller mellom 0 og m , og på den måten kan man telle antall a 'er som havner i samme intervall. Vi lar m være maksimal andel forekomst av den aktuelle TU-gruppen i et cellekjerne-bilde. Legg merke til at flate bilder vi gjøre at m blir veldig stor i forhold til hvor den maksimale grensen trenger å gå for å oppnå en god sannsynlighetsfordeling. Dette gjelder for de mest homogene TU-gruppene. Det er derfor viktig å passe på at ingen av bildene som blir analysert er tilnærmet helt flate. Av den grunn kan det være greit å se bort fra bilder som er for homogene. Fra tabell 4.3 kan vi observere at den høyeste andelen en gitt TU-gruppe opptar i bildet er 27 % i M98 og 50 % i L23, noe som betyr at det ikke blir nødvendig å se bort fra noen bilder.

Siden man har l intervaller vil størrelsen på et intervall være:

$$s = \frac{m}{l}$$

Verdien til et intervall, i_g , kan bli funnet på følgende måte:

$$h(i_g) = \frac{1}{m_i} \sum_{j=1}^{m_i} \Gamma_g(a_j)$$

hvor

$$\Gamma_g(a_j) = \begin{cases} 1 & \text{hvis } a_j \in [(g-1)s, gs] \\ 0 & \text{ellers} \end{cases}$$

Dermed kan man danne et histogram med intervallene i_g langs x-aksen, og $h(i_g)$ langs y-aksen. På denne måten får vi en sannsynlighetsfordeling for andelen forekomster av en gitt TU-gruppe som kan finnes i et bilde. Dette blir gjort for en hel gruppe, så vi kaller et slikt histogram et gruppehistogram. For å sette opp et tilsvarende histogram for en pasient går man frem på helt lik måte, men vi lar nå $h(i_g)$ kun bli basert på bildene til den aktuelle pasienten.

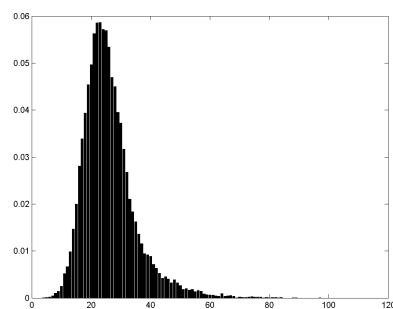
Når vi trener klassifikatoren ser vi på distansen en pasient har fra sin sannsynlighetsfordeling til de to prognosegruppenes sannsynlighetsfordelinger.

Dette kan medføre at en pasient i treningsettet får lavere avstand til sin egen prognosegruppe i forhold til hva pasienten i realiteten skulle hatt. Dette kommer av at pasienten er med på å forme sannsynlighetsfordelingen til prognosegruppen pasienten er med i. For å unngå dette kan man la være å ta med informasjon fra den aktuelle pasienten i sannsynlighetsfordelingen til prognosegruppen pasienten er medlem i, når man skal måle avstanden til denne pasienten. Vi kommer imidlertid ikke til å benytte oss av dette, og det er flere grunner til det. For det første blir det mindre data å danne estimatene til gruppenes sannsynlighetsfordelinger på, noe som kan gjøre at distansene vi måler fra en pasient til dennes prognosegruppe blir mindre generell enn det vi hadde håpet. For det andre blir det vanskeligere å analysere resultater basert på disse sannsynlighetsfordelingene ettersom det blir veldig mange ulike sannsynlighetsfordelinger å se på for en TU-verdi. For det tredje vil det å fjerne en pasient fra sannsynlighetsfordelingen til dens gruppe medføre at man ikke måler helt like avstander mellom pasienter som trener klassifikatoren og pasientene som blir brukt til test. Dette gjør analysen av resultatene etter test mer komplisert.

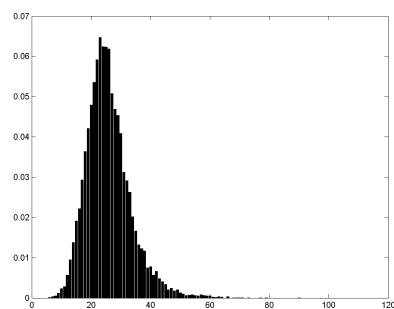
Selv om vi ikke kommer til å ta ut en pasient fra prognosegruppenes sannsynlighetsfordelinger når vi måler avstanden mellom gruppen og denne pasienten, ender vi fortsatt opp med en del sannsynlighetsfordelinger som må lages. Vi må nå lage et histogram for hver pasient, og et for hver prognosegruppe, for hver TU-gruppe. Disse kan vi bruke til å hente ut varians, som vi skrev om over, samt se om sannsynlighetsfordelingen er tilnærmet normalfordelt, eller om vi må ta hensyn til andre fordelinger. Et par eksempler på slike fordelinger er å finne i figur 4.6, og alle gruppehistogrammene kan ses i figurene 7.1 og 7.2 bakerst i Vedlegg A.

Som man kan se er det en stor likhet mellom prognosegruppene, men man kan for eksempel se at god prognose har høyere varians enn dårlig prognose. Man kan også legge merke til at de fleste histogrammer består av en topp, og er tilnærmet normalfordelte, med unntak av TU-gruppe 14, 15, 16 og 17. Det betyr at bruk av en distansemåler som antar normalfordeling, kan forsvares for de fleste TU-gruppene. Dette støtter derfor opp om bruk av Gaussisk Bhattacharyya distanse, men dette blir diskutert nærmere i seksjon 4.1.6.

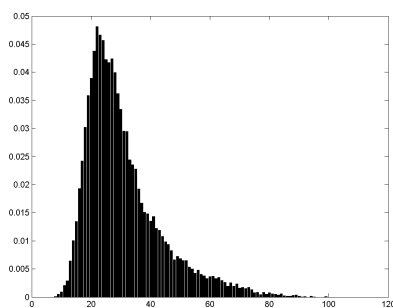
Det er verdt å merke seg at selv om histogrammene går fra 0 til 100, må man huske på at 100 kun representerer maks andel forekomst av den gitte TU-gruppen. Den maksimale andelen er gitt i tabell 4.3 for M98 og L23. I kolonnene finner vi TU-gruppen samt den maksimale prosentandelen denne TU-gruppen opptar i et bilde. Det vil si at man kan se på verdiene på x-aksen som prosentandel fra 0 til TU-gruppens prosentandel i denne tabellen. For eksempel vil plass 50 i histogrammet for TU-gruppe 3 fra M98 indikere andelen forekomster hvor $4.6\% \cdot 0.5 = 2.3\%$ av TU'ene til bildene er i TU-gruppe 3. Y-aksen i histogrammene indikerer det normaliserte antallet forekomster av sannsynlighetene for TU-gruppene.



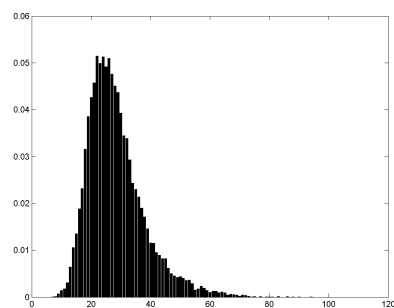
(a) TU-gruppe 1, god prognose



(b) TU-gruppe 1, dårlig prognose



(c) TU-gruppe 11, god prognose



(d) TU-gruppe 11, dårlig prognose

Figur 4.6: Noen sannsynlighetsfordelinger for begge prognosegruppene uten ekskludering av pasienter fra M98.

Til slutt er det verdt å legge merke til at de siste TU-gruppene ikke forekommer i alle TU-bilder. Fra tabell 4.3 ser vi at enkelte TU-gruppers maksimale andel forekomster i bildet er veldig lav. Dette gjelder spesielt TU-gruppe 14 og 17 for M98, som maksimalt opptar i underkant av 5 % av bildet. For L23 gjelder dette kun TU-gruppe 17. Det at enkelte TU-grupper ikke forekommer i alle TU-bilder gjør at mange sannsynlighetsfordelinger ikke blir like godt representert ved normalfordeling som hvis man ser bort fra $h(0)$. Skulle man ønske å benytte en distansemåler som antar normalfordeling, bør man se bort fra andelen forekomster av 0.

Tabell 4.3: Maksimal andel forekomster i prosent av hver TU-gruppe. Dataene er hentet fra pasienter i henholdsvis M98 og L23.

TU-gruppe	1	2	3	4	5	6	7	8	9
M98	4	5	5	5	4	5	11	14	27
L23	7	9	10	12	9	11	20	28	50

TU-gruppe	10	11	12	13	14	15	16	17
M98	4	6	3	4	< 1	1	2	< 1
L23	5	12	5	8	1	1	3	< 1

4.1.4 Varians

Som vi har vært inne på, har det i flere tilfeller vist seg at det å kombinere LBP med varians gir bedre klassifiserings resultater enn ved å bruke LBP alene [54] [36]. Av den grunn velger vi å beregne variansen for hver av de pikslene som blir brukt som senter-pikslar i beregningen av LBP, slik at vi kan benytte varians som en tilleggs-egenskap til LBP. Variansen blir beregnet på følgende måte:

$$VAR = \frac{1}{P-1} \sum_{p=1}^P (g_p - \bar{g})^2$$

hvor $\bar{g} = \frac{1}{P} \sum_{p=1}^P g_p$, g_p er en piksel i nabolaget og P er størrelsen på nabolaget. For å få med informasjon angående gråtonen til senter-pikselen, normaliserer vi variansen med denne gråtonen, og oppnår følgende operator:

$$VAR_c = \frac{1}{(P-1)g_c} \sum_{p=1}^P (g_p - \bar{g})^2$$

hvor g_c er gråtonen til senter-pikselen. Ettersom gråtonene i cellekjernebildene i både M98 og L23 ligger i intervallet $[0 \ 1023]$, vil man kunne støte på tilfeller hvor $g_c = 0$. I vårt tilfelle forekommer dette veldig sjeldent, men når det forekommer deler vi på 1 og ikke 0.

4.1.5 Kvantisering av varians

VAR_c er en kontinuerlig funksjon, og vi er derfor nødt til å kvantisere VAR_c -verdiene. Ettersom bildene vi jobber med består av 1024 gråtoner, kan vi i verste fall ende opp med en minste verdi lik 0 og en maks verdi på

$$\frac{4(1023 - 511.5)^2 + 4(0 - 511.5)^2}{7} \approx 299008$$

med et 8 piksels stort nabolag. Dette utgjør et veldig stort intervall, og vil være uhåndterlig. Av den grunn er vi nødt til å kvantisere disse verdiene. Det første vi må merke oss er at slike høye VAR_c -verdier forekommer svært sjeldent. Når man i tillegg deler på gråtone-verdien til senter-pikselen viser det seg at svært få cellekjerne-bilder har forekomster av VAR_c høyere enn 255. For eksempel er det ingen slike forekomster i noen av cellekjerne-bildene til pasient M98-020. Ved å si at alle VAR_c -verdier større eller lik 255 blir satt til 255 og resten av verdiene blir rundet av til nærmeste heltall, kan vi lagre alle VAR_c -verdiene som bilder med 256 gråtoner på samme måte som for LBP. Vi er imidlertid nødt til å kvantisere enda mer før vi bruker disse verdiene, ettersom 256 fortsatt er et uhåndterlig tall i forhold til vårt formål med VAR_c .

En mulighet er å lage N like store bins mellom 0 og 255, men dette kan resultere i flere nesten tomme bins ettersom de ulike VAR_c -verdiene kan forventes å være relativt lave. Dette kan vi forvente på grunn av normaliseringen vi utfører med hensyn på senter-pikselen, og fordi det tilsynelatende

ikke er veldig store variasjoner i såpass lokale områder i cellekjerne-bildene. For å danne en oversikt over hvilke verdier for VAR_c vi står overfor, setter vi opp et histogram som viser andelen forekomster av de ulike verdiene. I figur 4.7 kan man se andelen forekomster av de 256 VAR_c -verdiene basert på pasientene i treningsettet i M98. Andelen forekomster basert på pasientene i treningsettet i L23 er tilnærmet lik, hvilket kommer frem i figur 4.8. Som vi kan se i histogram 4.7a, finner vi de fleste av verdiene mellom 0 og 100. Når man derimot kun ser på den første delen av histogram 4.7b kan man se at forekomstene avtar fort, og blir svært lave fra VAR_c -verdi 60 og ut. Vi begrenser oss til 10 grupperinger av VAR_c -verdiene, og lar det 10. kvantiseringsnivået gå fra 60 til maks VAR_c -verdi. Videre velger vi følgende kvantisering, hvor venstresiden tilsvarer faktisk VAR_c , og høyresiden er gruppen verdiene blir kvantisert til:

$$1 - 2 \rightarrow 1$$

$$3 - 4 \rightarrow 2$$

$$5 - 6 \rightarrow 3$$

$$7 - 9 \rightarrow 4$$

$$10 - 13 \rightarrow 5$$

$$14 - 19 \rightarrow 6$$

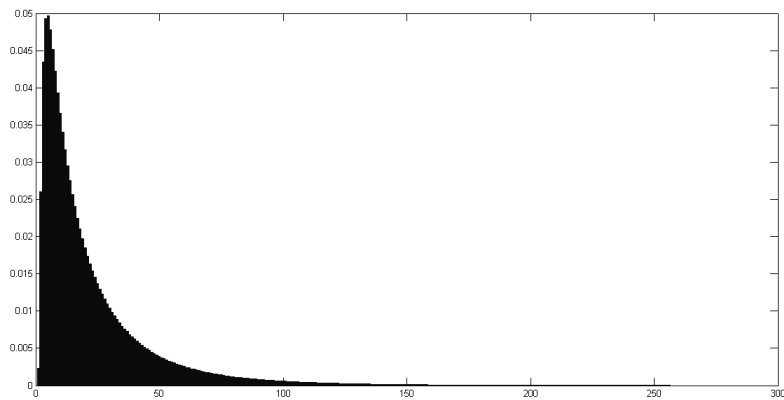
$$20 - 27 \rightarrow 7$$

$$28 - 37 \rightarrow 8$$

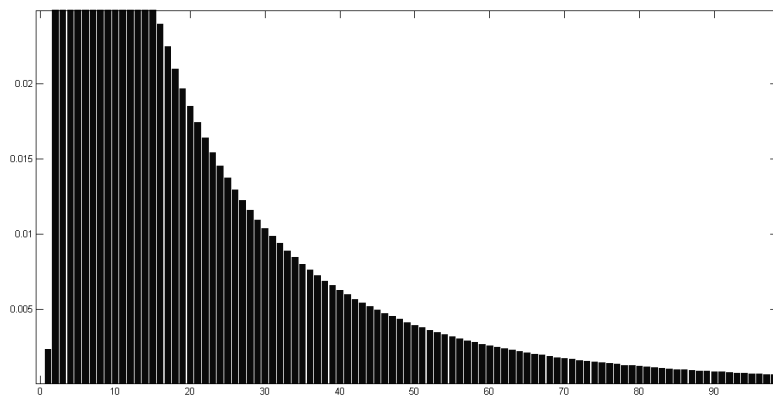
$$38 - 59 \rightarrow 9$$

I figur 4.8 kan man se to fordelinger av VAR_c -gruppene, en basert på treningsettet i M98 og en på treningsettet i L23. Her kan man legge merke til at verdiene er forholdsvis jevnt fordelt, og at fordelingene i M98 og L23 er veldig like. Kvantiseringen er gjort ved å se på figur 4.7a og 4.7b, og ut fra disse har vi manuelt valgt hvordan fordelingene av de ulike VAR_c -gruppene skal bli oppdelt. Kvantiseringen er gjort på grunnlag av dataene fra treningssettet, og det er denne kvantiseringen vi bruker videre i oppgaven. Her kunne vi benyttet oss av Lloyd-Max kvantisering [23], for å gjøre kvantiseringen optimal. Vi velger imidlertid å anta at vår kvantisering er tilstrekkelig, og av den grunn benytter vi oss ikke av Lloyd-Max kvantisering her.

Når VAR_c er kvantisert setter vi opp sannsynlighetsfordelinger for VAR_c på tilsvarende måte som for TU, se seksjon 4.1.3. Disse sannsynlighetsfordelingene kan nå brukes på tilsvarende måte som sannsynlighetsfordelingene til TU-gruppene. På denne måten kan ender vi opp med distanser fra både VAR_c -grupper og TU-grupper, som igjen kan brukes som egenskaper i klassifisering.

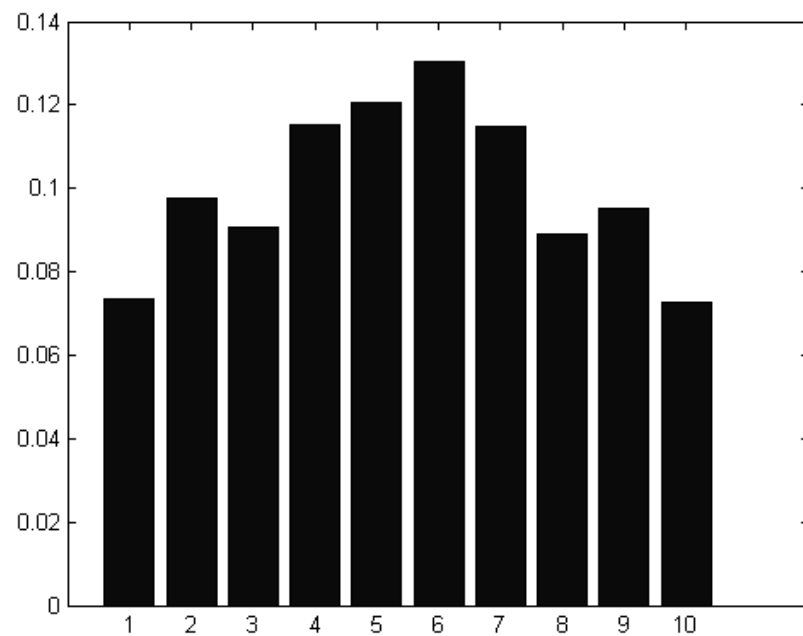


(a) Hele histogrammet

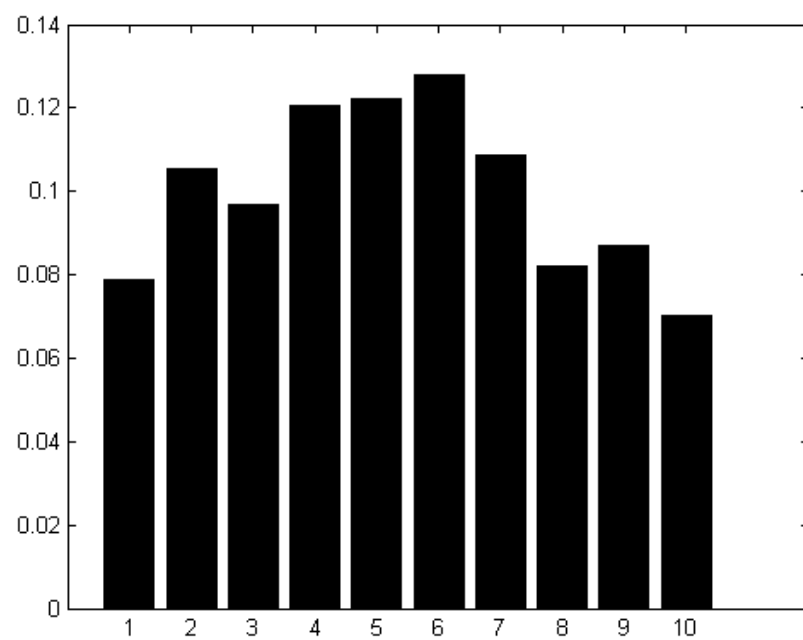


(b) De 100 første VAR_c -verdiene i histogrammet.

Figur 4.7: Denne figuren viser en oversikt over hvilke verdier for VAR_c som forekommer i 8 piksels nabolag i cellekjerne-bildene i god prognose i treningsettet i M98. Her er alle VAR_c -verdier blitt lagt til 1. Dette er gjort for å gjøre verdiene enklere å jobbe med i MATLAB.



(a) Fordelingen av VAR_c -gruppene i M98.



(b) Fordelingen av VAR_c -gruppene i L23.

Figur 4.8: Denne figuren viser fordelingen av VAR_c -gruppene i M98 og L23.

4.1.6 Valg av distansemåler for TU

Ved å studere histogrammene over sannsynlighetsfordelingene for alle TU- og alle VAR_c -grupper, kan man finne ut hvilken distansemåler det vil være lurt å bruke til klassifisering. De fleste histogrammene over sannsynlighetsfordelingene for TU-gruppene ser ut til å kunne tilpasses en normalfordeling. Man kan også finne noen histogrammer som kan tilpasses Chi-kvadratfordelingen, men det er betydelig mindre av disse histogrammene. Det vil derfor kunne lønne seg å bruke en distansemåler som antar normalfordeling. Av de tidligere omtalte distansemålerne, er Mahalanobis og Gaussisk Bhattacharyya distansene tilpasset normalfordelingen. Dersom man ser på forventningsverdien til de to forskjellige klassene for hver TU-gruppe, ser man at de er ganske like. Man kan derimot se at det er større forskjell i variansen til de to klassene for hver TU-gruppe. Det vil derfor være mer hensiktsmessig å ta i bruk et distansemål som i høyere grad tar hensyn til varians enn forventningsverdi. Av den grunn velger vi å gå videre med Gaussisk Bhattacharyya distanse, gitt ved

$$\frac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \left(\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2} \right)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2} \ln \frac{|\frac{1}{2}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)|}{|\boldsymbol{\Sigma}_1|^{1/2} |\boldsymbol{\Sigma}_2|^{1/2}}$$

hvor $\boldsymbol{\mu}_i$ og $\boldsymbol{\Sigma}_i$ er henholdsvis forventningsverdivektoren og kovariansmatrisen til klasse ω_i .

4.1.7 Klassifikator

Da vi skrev om valg av distansemåler, så vi at forventningsverdiene til de to klassene er veldig like. På grunnlag av dette vil ikke bruk av minimum distanse klassifisering være et godt valg. Fra spredningsplottene i figur 4.9 kan man se at dataene til de to klassene ligger blandet. Av denne grunn velger vi å heller ikke bruke Nærmeste nabo klassifisering.

Som vi var inne på tidligere kan mange naturlige fenomener representeres ved normalfordeling. Vi antar derfor at hver klasse kan tilnærmes med en Gaussisk sannsynlighetsfordeling. Av den grunn velger vi å benytte multi-variabel Gaussisk sannsynlighetstetthetsfunksjon [23] gitt ved

$$p(\mathbf{x}|\omega_s) = \frac{1}{(2\pi)^{\frac{d}{2}} |\boldsymbol{\Sigma}_s|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_s)^t \boldsymbol{\Sigma}_s^{-1} (\mathbf{x}-\boldsymbol{\mu}_s)}$$

hvor \mathbf{x} er egenskapsvektoren til en pasient, ω_s er en klasse, $\boldsymbol{\mu}$ er en vektor som inneholder forventningsverdiene til hver egenskap for den gitte klassen og $\boldsymbol{\Sigma}$ er kovariansmatrisen til den samme klassen. For å klassifisere pasientene er vi imidlertid nødt til å finne $p(\omega_s|\mathbf{x})$. Fra Bayes teorem [17] vet vi at

$$p(\omega_s|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_s)p(\omega_s)}{p(\mathbf{x})}$$

Siden det er like viktig å klassifisere pasienter med god og dårlig prognose riktig, settes $p(\omega_0) = p(\omega_1) = 0.5$. $p(\omega_s|\mathbf{x})$ blir beregnet for begge klassene,

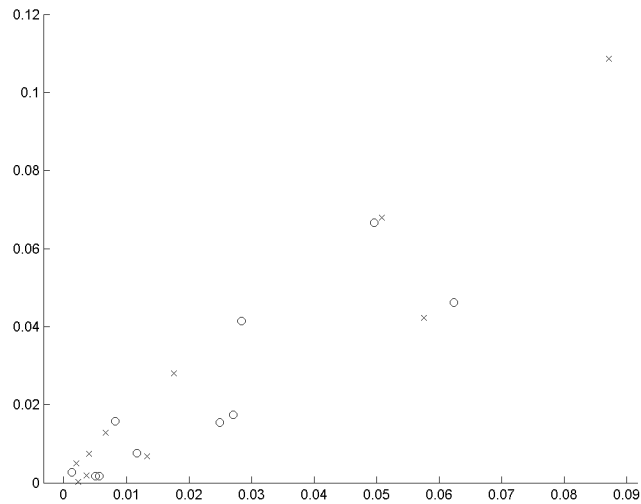
og pasienten blir klassifisert til den klassen med høyest sannsynlighet. Med andre ord velges klasse ω_s hvis og bare hvis $p(\omega_s|\mathbf{x}) \geq p(\omega_t|\mathbf{x})$ for alle $t \neq s$.

4.1.8 Kort om visualisering av klassene

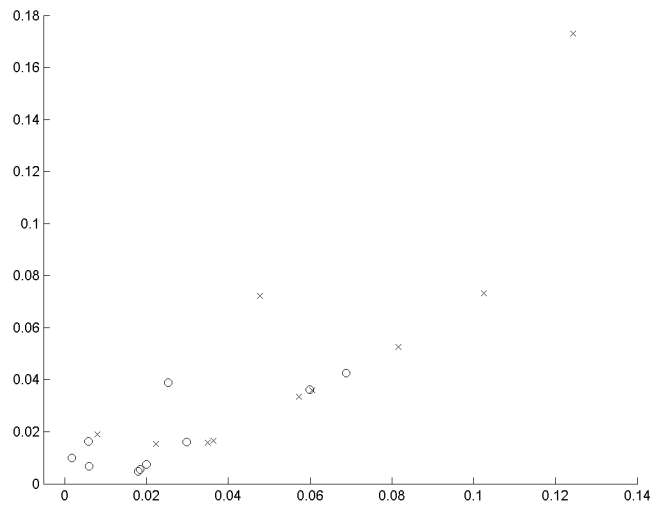
I figur 4.9 ser man to spredningsplott som viser avstanden beregnet fra hver pasients sannsynlighetsfordeling til de to prognosegruppens sannsynlighetsfordelinger, for TU-gruppe 5 og 12. Slike plott kommer det flere av i oppgaven, og grunnen til at vi har valgt å visualisere dette på denne måten er fordi hver TU-gruppe (og returverdier fra andre operasjoner) har to distanser. Når vi jobber med en TU-gruppe (eller returverdier fra andre operasjoner) bruker vi begge distansene som en pasients egenskap. Av den grunn viser slike plott godt hvordan pasientene i de to prognosegruppene forholder seg i forhold til hverandre med distanser beregnet fra en gitt operator. Selv om en klassifikator viser korrekt andel klassifisering, vil man ikke kunne se hvor ulike/like pasientenes distanser er i prognosegruppene. Det kan vi derimot med spredningsplott som det i figur 4.9, og vi kan på den måten observere hvor langt unna pasienter er eventuelle skiller mellom god og dårlig prognose, og på den måten visualisere robusthet i metoder.

4.1.9 Valg av distansemaal for varians

Når man ser på histogrammene over sannsynlighetsfordelingene for alle VAR_c -gruppene (figur 7.4 og 7.3 i Vedlegg A), kan man legge merke til at det er en betydelig større andel histogrammer som kan tilpasses Chikvadratfordelingen, i forhold til histogrammene for TU-gruppene. Det er riktignok slik at det også her er en større andel som kan tilpasses normalfordelingen, så det ser ut til at det også her vil bli best å bruke et distansemaal som er tilpasset normalfordelingen. Fra sannsynlighetsfordelingene funnet ved bruk av VAR_c -gruppene i treningssettet i M98 kan man, i likhet med for TU-gruppene, observere at forventningsverdiene for de to klassene er omtrent den samme. Der er derimot større variasjon i standardavviket til de to klassene. Av den grunn vil det også her være hensiktsmessig å velge Gaussisk Bhattacharyya.



(a) Sannsynlighetsfordelingene er basert på TU-gruppe 5.



(b) Sannsynlighetsfordelingene er basert på TU-gruppe 12.

Figur 4.9: Spredningsplottene viser de 20 pasientene i treningssettet fra M98. x -aksen viser en pasients distanse til klassen med god prognose, mens y -aksen viser pasientens distanse til klassen med dårlig prognose. Distansene er funnet ved å beregne Gaussisk Bhattacharyya avstand mellom sannsynlighetsfordelingene.

4.2 LBP benyttet på nabolag bestående av 16 piksler

Ojala et al. [54] oppnår bedre klassifiseringsresultater ved å bruke et større nabolag enn et nabolag med 8 naboer og en radius på 1. Vi starter i likhet med tidligere repeterer vi den generelle formelen for LBP, gitt ved

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p$$

hvor P er antall naboer og R er radiusen. Dersom man ønsker et 16 piksels stort nabolag med radius 2, ender man opp med 4116 rotasjonsinvariante binære mønstre, eller TU-verdier. Dette er et veldig stort antall egenskaper, og av den grunn ser vi først på muligheten til å gruppere TU-verdiene etter samme mønster som når vi har et nabolag på 8 piksler og en radius på 1, se seksjon 4.1.2. Ved å følge denne fremgangsmåten ender vi opp med 119 TU-grupper, hvilket er et for høyt antall å jobbe med. Dersom en følger grupperingen til Ojala et al. [54] tar man vare på de uniforme mønstrene, slik at de havner i hver sin gruppe. De resterende mønstrene blir slått sammen til en gruppe, slik at man ender opp med 18 TU-grupper. Som vi var inne på i seksjon 4.1 vil vi miste en del informasjon fra de ikke-uniforme mønstrene. Andre grupperinger, som den vi bestemte oss for ved bruk av $LBP_{8,1}$ og den til Ma [36] (30 TU-grupper), blir fort mer krevende å jobbe med.

Ved beregning av varians benytter vi oss av tilsvarende formel som i seksjon 4.1.4. For et 16 piksels nabolag er denne gitt ved

$$VAR_c = \frac{1}{15g_c} \sum_{p=1}^{16} (g_p - \bar{g})^2$$

hvor g_c er gråtonen til senter-pikselen, og g_p er funnet i tilsvarende sirkel som pikslene ble funnet for $LBP_{16,2}$. Foruten disse endringene, vil fremgangsmåten være lik som i seksjon 4.1.

4.3 LBP benyttet på nabolag bestående av 8 piksler i nedskalerte cellekjerne-bilder

Som vi var inne på i seksjon 2.3 kan man enten øke størrelsen på nabolaget av piksler man anvender operatorene på, eller man kan skalere ned det originale bildet før man benytter operatorene på de nedskalerte bildene. I denne seksjonen tar vi for oss den siste av disse to fremgangsmåtene. Siden vi ikke øker nabolaget til operatorene for denne fremgangsmåten, kan vi benytte de samme operatorene som i seksjon 4.1 og 4.1.4. Selv om vi er nødt til å bruke tid på å nedskalere hvert cellekjerne-bilde før operatorene benyttes på disse, vil ikke beregningstiden nødvendigvis øke. Dette kommer av at operatorene kun blir benyttet på 1/4 av pikslene som blir brukt av tilsvarende operatører på de originale bildene.

4.3.1 Interpolasjon

I seksjon 2.3.1 skrev vi at det må tas hensyn til kvaliteten av tilnærmingen og tidsbruken ved valg av interpolasjonsmetode. I denne oppgaven er det viktig å ta vare på den lokale teksturen, og den interpolasjonsmetoden som gjør dette best er bikubisk interpolasjon. Denne metoden er også den mest regnekrevende interpolasjonsmetoden, men vi kan i denne oppgaven tillate oss å bruke en litt mer regnekrevende operator. Dette kommer av at vi ikke er avhengig av et resultat innen en veldig kort tid, selvom vi selvfølgelig ønsker at det totale resultatet skal fremkomme innenfor rimelige tidsrammer. I tillegg sparer vi som sagt inn tid på beregning av TU, grunnet færre piksler i bildet. På grunnlag av dette velger vi å bruke bikubisk interpolasjon i nedskaleringen.

4.3.2 Skalering

I seksjon 2.3.2, som omhandlet bildepyramider, skrev vi at det i praksis ikke er noen grunn til å benytte LBP på veldig små cellekjerne-bilder. Dette kommer av at cellekjerne-bilder som er nedskalert med en forholdsvis høy faktor vil ha mistet veldig mye tekstur-informasjon, og på den måten blitt mer eller mindre ubrukelige til vårt formål. I tillegg vil for små bilder resultere i et lite utvalg av TU, noe som kan medføre svært unøyaktige estimerer til sannsynlighetsfordelingene vi senere bruker for å hente ut egenskaper til klassifisering. Vi vet fra tidligere at gjennomsnittlig størrelse på originalbildene i M98 er ca 95×95 piksler, hvor en cellekjerne opptar ca. 33 % av bildet. Med en nedskaleringsfaktor på 2 ender vi opp med bilder med størrelser rundt 48×48 piksler, noe som betyr at selve cellekjernen opptar ca 760 piksler. Dersom man nedskalerer med en faktor på 4 vil cellekjernen oppta ca 190 piksler, og med en faktor på 8 vil cellekjernen oppta ca. 48 piksler og den tilhørende radiusen vil være ca 4. Ettersom det ikke blir beregnet en TU-verdi for pikslene langs omkretsen av cellekjernene, vil vi med en nedskaleringsfaktor på 4 sitte igjen med omtrent 140 TU, mens med en faktor på 8 blir antallet TU omtrent 28. Dette er helt klart ikke nok data til å estimere sannsynlighetsfordelingene til de 17 TU-gruppene.

Cellekjerne-bildene i L23 er gjennomsnittlig 80×80 piksler store, hvor cellekjernen opptar ca 60 % av hele bildet. Dette gjør at en gjennomsnittlig cellekjerne i L23 består av ca 3800 piksler, mens en gjennomsnittlig cellekjerne i M98 består av ca 3000 piksler. Selv om et gjennomsnittlig cellekjerne-bilde i L23 består av flere piksler enn i M98 vil det ikke være hensiktsmessig å nedskalere cellekjernene med en faktor på 4 eller mer. Vi kan for eksempel se at en faktor på 4 ville gjort at en cellekjerne i L23 hadde bestått av gjennomsnittlig 236 piksler, og som vi var inne på for M98 vil ikke dette antallet være tilstrekkelig for vårt formål. Av den grunn kommer vi kun til å benytte nedskalering med en faktor på 2 i begge datasettene. Etter at bildene er nedskalert er fremgangsmåten lik som i seksjon 4.1.

4.4 Adaptive egenskaper

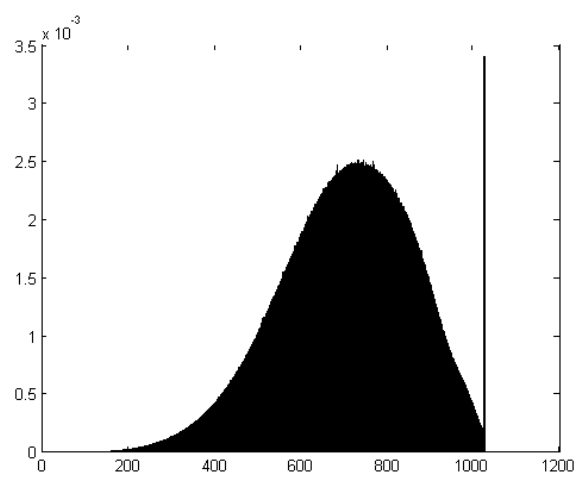
I denne seksjonen tar vi for oss tilpasningsdyktige (adaptive) egenskaper, som vi skrev om i seksjon 2.1.1. I stedet for å se på mange ulike kombinasjoner, slik vi har gjort i de tidligere seksjonene, kan man benytte egenskaper som tilpasser seg det gitte bildematerialet. I denne seksjonen vurderer vi tre ulike kombinasjoner av operatører, før vi lager matriser basert på forekomstene av de ulike verdiene fra disse operatorene. Deretter ser vi hvor i disse matrisene det ser ut til å ligge mest klasseskillende informasjon. Dette gjøres ved å bruke fremgangsmåten Nielsen et al. benyttet i [45], ved bruk av adaptive egenskaper. Etter at egenskapene er funnet, bruker vi dem i forsøk på å skille pasienter med god og dårlig prognose.

Som vi så i seksjon 2.1.7 kom Ojala et al. [54] og Ma [36] frem til at LBP kombinert med andre operatører, som for eksempel varians, ga bedre resultater enn LBP alene. Tidligere har vi sett på andelen forekomster av TU og varians, men i den fremgangsmåten så vi på sannsynlighetsfordelingene til TU- og VAR_c -gruppene hver for seg. Denne fremgangsmåten resulterer i 1D-histogrammer som representerer sannsynlighetsfordelingene til TU og VAR_c . Et alternativ til dette kan være å se på hvor ofte en senterpiksel, c , har naboer som resulterer i både en gitt TU-gruppe og en gitt VAR_c -gruppe. Disse forekomstene kan vi sette inn i en matrise, i stedet for en vektor. Selv om vi ønsker å la den ene forekomsten bli målt basert på TU, trenger ikke den andre operatoren å være VAR_c . I stedet kunne vi, for eksempel, sett på gråtonen til senterpikselen eller den lokale entropien rundt senterpikselen.

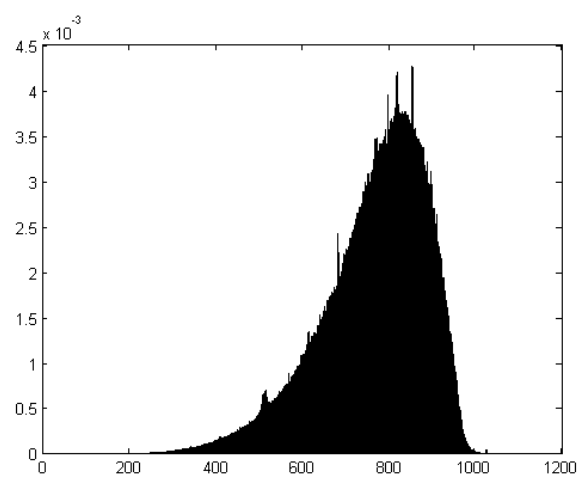
For å danne en slik matrise, M_T , kan man la $M_T(\tau, \eta)$ være andelen forekomster av tilfeller hvor en senterpiksel har nabolag som gir TU-gruppe τ og en verdi fra den andre operatoren som vi kaller η . Vi kommer nå til å ta for oss tre tilfeller av hva η kan være, før vi går videre med hvordan vi skal bruke M_T til klassifisering.

4.4.1 Kombinasjon mellom senterpiksels gråtone og nabolagets TU-gruppe

Den første matrisen vi tar for oss viser TU-gruppe mot gråtone-verdien til senter-pikselen, c , og vi kaller denne matrisen M_{TG} . Som vi har skrevet tidligere er LBP gråtoneinvariant, men vi har ingen indikasjoner på om bruk av gråtoneinvariante metoder er bedre ved analyse av cellekjerne-bilder. Vi har til nå sett på metoder som analyserer en TU-gruppe alene, men vi har ikke sett på andel forekomster av en TU i forhold til gråtonen senterpikselen har. Denne fremgangsmåten kan derfor føre til at vi finner egenskaper som skiller klassene bedre enn det TU gjør alene. Vi starter med å se på senterpiksels gråtone fordi gråtoner i cellekjerne-bildene baseres på hvor mye kromatin det finnes i et område. For eksempel kan en kromatin-ansamling bli sett som et mørkt område i cellekjernen. I [32] blir det nevnt at forandringer i kompakt kromatin i cellene blir sett på som en viktig parameter i patologisk klassifisering, blant annet når man skal stille diagnose.



(a) M98



(b) L23

Figur 4.10: Fordelingen av gråtoner over bildene til alle pasientene i treningsettet i henholdsvis M98 og L23. Histogrammene er normalisert.

Gråtonen c ligger i intervallet $[0, 1023]$, noe som resulterer i en for stor matrise hvis vi beholder full gråtoneoppløsning. Av den grunn velger vi å utføre en kvantisering av antall gråtoner. For å gjøre dette ser vi på andelen forekomster av de ulike gråtonene. Vi setter derfor opp et histogram basert på pasientene i treningsettet i M98 som inneholder disse andelene. Dette histogrammet er å finne i figur 4.10, sammen med et tilsvarende basert på treningsettet i L23. Som vi kan se er formen på figurene tilnærmet normalfordelte, men vi kan også se en høy søyle på plass 1023 for M98, noe som tyder på at alle verdiene etter denne har blitt flyttet hit for å begrense antall gråtoner til $2^{10} = 1024$. Vi kvantiserer gråtonene i forhold til formen for å oppnå gråtone-grupperinger med tilnærmet like mange gråtoner. I tillegg lar vi antall grupperinger bli begrenset til 10 for ikke å få for mange grupper å jobbe med. Ved kvantiseringen har vi gått frem på liknende måte som for kvantisering av varians, se seksjon 4.1.5. Forskjellen er at kvantiseringen er gjort automatisk i et dataprogram, ved å sørge for at fordelingen blir mest mulig flat. Kvantiseringen tar hensyn til den tilnærmet logaritmiske formen til histogrammene, og av den grunn ender vi opp med følgende kvantisering:

$$0 - 496 \rightarrow 1$$

$$497 - 572 \rightarrow 2$$

$$573 - 626 \rightarrow 3$$

$$627 - 672 \rightarrow 4$$

$$673 - 714 \rightarrow 5$$

$$715 - 755 \rightarrow 6$$

$$756 - 797 \rightarrow 7$$

$$798 - 843 \rightarrow 8$$

$$844 - 901 \rightarrow 9$$

$$902 - 1023 \rightarrow 10$$

Med kvantiseringen på plass, vil T_{MG} bli en 17×10 stor matrise.

4.4.2 Kombinasjon mellom variansen i det lokale nabolaget og nabolagets TU-gruppe

Som vi har vært inne på tidligere, har TU kombinert med varians vist seg å gjøre det godt i blant annet [36] og [54]. Det er derfor naturlig å se på en matrise, M_{TV} , som ser på andelen forekomster av en TU-gruppe kombinert med varians. Med andre ord vil en plass i matrisen $M_{TV}(\tau, \eta)$ betegne andelen forekomster av TU-gruppe τ , samt varians-gruppe η . Før vi kan benytte dette er vi imidlertid, som med tilfellet hvor η var gråtonen til senterpiksel, nødt til å gå gjennom hvordan vi finner variansen, samt utføre en tilhørende kvantisering.

Tidligere i dette kapittelet så vi på muligheten å kombinere TU med VAR_c , gitt ved

$$VAR_c = \frac{1}{7g_c} \sum_{i=1}^8 (g_i - g_c)^2$$

hvor g_i er en nabopiksel til senterpikselen g_c . Når vi nå skal kombinere TU og varians ser vi ingen grunn til å ikke bruke denne typen variansmåling. Dette vil også gi oss et bedre sammenlikningsgrunnlag mellom de ulike variantene av kombinerings. Vi har allerede vært inne på hvordan vi kvantiserer utverdiene til VAR_c , i seksjon 4.1.4, så vi kommer ikke til å ta for oss dette her. Av den grunn ender vi opp med en matrise, M_{TV} , som er 17×10 stor.

4.4.3 Kombinasjon mellom entropien i det lokale nabolaget og nabolagets TU-gruppe

I [32] kom Jørgensen et al. frem til at bruken av entropi benyttet på cellekjerne-bilder i prostata førte til gode resultater. Entropi kombinert med andre egenskaper viste seg videre å bli svært gode, og av den grunn kan det være interessant å se om entropi gjør seg bra som en egenskap i kombinasjon med LBP. Det finnes flere måter å måle entropien, og her velger vi å bruke Shannon entropi, som er gitt ved

$$ENT = - \sum_{i=0}^{n-1} p(g_i) \log_2(p(g_i))$$

hvor n er antall ulike gråtoner i nabolaget man ser på, $p(g_i)$ er sannsynligheten for en gråtone g_i i nabolaget og \log_2 er 2-logaritmen. Dette er den samme formelen for entropi som Jørgensen et al. [32] benyttet seg av. For å gjøre bruken av entropi konsistent med bruken av $LBP_{8,1}$ og $VAR_{c8,1}$, lar vi nabolaget entropien baseres på være tilsvarende som for disse. Hvis man tenker seg at det er n ulike gråtoner i nabolaget, og sannsynligheten for en gråtone er $p(g_i) = \frac{n_i}{8}$ blir

$$ENT_{8,1} = \sum_{i=0}^{n-1} \frac{n_i}{8} \log_2 \left(\frac{n_i}{8} \right)$$

hvor $0 < n \leq 8$ og $\sum_{i=0}^{n-1} n_i = 8$. Ved å bruke denne operatoren observerer vi at antallet ulike entropi-verdier er 21. Dette kan man blant annet se i tabell 4.4. Hvis vi skal finne andelen forekomster av de 21 entropi-verdiene, for deretter å behandle hver av disse forekomstene uavhengig av hverandre, ser vi at vi vil få en ikke-lineær organisering av entropi-verdiene. Selv om 21 er et overkommelig antall verdier å jobbe med, ønsker vi å holde oss til en lineær fordeling av entropi-verdiene. Vi velger derfor å kvantisere verdiene til 10 entropi-grupper, hvor kvantiseringen er lineær. Denne kvantiseringen er å finne i tabell 4.5.

Tabell 4.4: Tabellen viser entropien til de ulike tilfellene man står overfor når man regner ut entropien i det lokale området. 2 forekomster er merket med fete bokstaver for å illustrere at entropien som blir oppnådd, er oppnådd tidligere i tabellen. Med andre ord ser vi at det er 21 ulike entropi-verdier som kan bli funnet med $ENT_{8,1}$.

Nr.	Antall like gråtoner	Entropi
1	8	0
2	7 + 1	0,544
3	6 + 2	0,811
4	6 + 1 + 1	1,061
5	5 + 3	0,954
6	5 + 2 + 1	1,299
7	5 + 1 + 1 + 1	1,549
8	4 + 4	1,000
9	4 + 3 + 1	1,406
10	4 + 2 + 2	1,500
11	4 + 2 + 1 + 1	1,750
12	4 + 1 + 1 + 1 + 1	2,000
13	3 + 3 + 2	1,561
14	3 + 3 + 1 + 1	1,811
15	3 + 2 + 2 + 1	1,906
16	3 + 2 + 1 + 1 + 1	2,156
17	3 + 1 + 1 + 1 + 1 + 1	2,406
12	2 + 2 + 2 + 2	2,000
18	2 + 2 + 2 + 1 + 1	2,250
19	2 + 2 + 1 + 1 + 1 + 1	2,500
20	2 + 1 + 1 + 1 + 1 + 1 + 1	2,750
21	1 + 1 + 1 + 1 + 1 + 1 + 1 + 1	3,000

Tabell 4.5: Tabellen viser kvantiseringen av entropi-verdiene. Her representerer Entropi-gr. den nye inndelingen og Entropi-nr. viser hvilke entropier som havner i en gruppe. Entropi-numrene er i samsvar med numrene i tabell 4.4.

Entropi-gr.:	Entropi-intervall:	Entropi-nr.:
1	[0, 0.3]	1
2	(0.3, 0.6]	2
3	(0.6, 0.9]	3
4	(0.9, 1.2]	4, 5, 8
5	(1.2, 1.5]	6, 9, 10
6	(1.5, 1.8]	7, 11, 13
7	(1.8, 2.1]	12, 14, 15
8	(2.1, 2.4]	16, 18
9	(2.4, 2.7]	17, 19
10	(2.7, 3.0]	20, 21

4.4.4 Valg av kombinasjon

Til nå har vi beskrevet tre ulike operatører det kan være aktuelt å kombinere med $LBP_{8,1}$. For å finne ut hvilke av de tre operatorene vi skal kombinere med $LBP_{8,1}$, ser vi hvor godt disse skiller god og dårlig prognose sammen med TU. Dette gjør vi ved å sette opp spredningsplott med TU langs x-aksen og η langs y-aksen. Deretter er gjennomsnittlig TU-gruppe og η for hver cellekerne plottet. Dette resulterer i mange elementer i hvert plott og plottene blir noe uoversiktlige. Skulle vi plottet gjennomsnittet til hver pasient, ville vi imidlertid mistet mye informasjon, noe som kunne medført svært unøyaktige antakelser om de ulike operatorene. Man kan fortsatt si at vi mister en del informasjon ved å ta gjennomsnittet til hvert cellekerne-bilde, men hadde vi sett på alle forekomster av alle TU kombinert med η , ville vi ikke kunne sett noe klasseskillende informasjon i plottene. Dette kommer av at man kan forvente at alle TU-grupper kombinert med alle η vil forekomme i minst noen tilfeller uavhengig av prognosegruppe.

De tre spredningsplottene, ett for hver operator, er å finne i figur 4.11. I figuren observerer vi mye overlapp mellom god og dårlig prognose, men vi kan i samtlige plott se at cellekerne-bildene fra pasienter med god prognose har litt mer varierende gjennomsnitt i både TU og η . Figur 4.12 gir et bedre inntrykk av hvor differansene mellom god og dårlig prognose befinner seg, samt hvor stor differansen er. Det er tilsynelatende størst differanse mellom god og dårlig prognose ved bruk av entropi. Av den grunn velger å basere adaptive egenskaper på kombinasjon mellom TU og entropi.

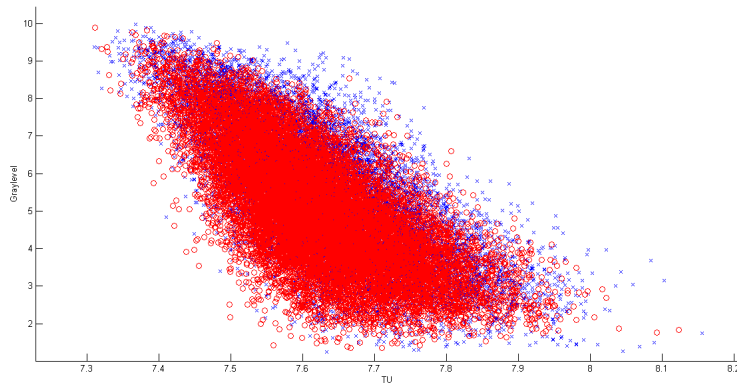
I de tidligere seksjonene har vi sett på kombinasjoner mellom TU- og VAR_c -grupper. For å danne et best mulig sammenlikningsgrunnlag mellom bruk av klassiske og adaptive metoder, kan det være et poeng i å også danne adaptive egenskaper basert på TU- og VAR_c -grupper. Vi vet at det er en sammenheng mellom varians og entropi [65]. Shannons entropi-ulikhet er definert som

$$ENT(\varphi) \leq \log_2 \sqrt{2\pi e VAR(\varphi)} = \log_2(\sigma) + \log_2(\sqrt{2\pi e})$$

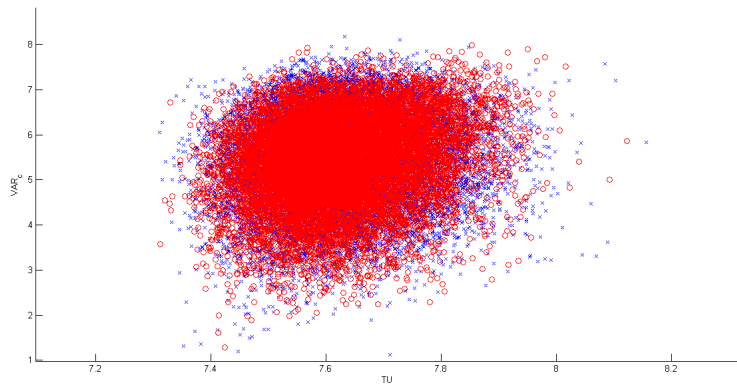
hvor ENT er Shannon entropi, VAR er varians, φ er en sannsynlighetsfordeling og σ er standardavvik, og ulikheten blir en likhet når fordelingen er en (sum av) Gauss fordeling(er). Vi ser at entropi og varians har en ikke-lineær sammenheng, hvilket innebærer at sammenlikning mellom tidligere kombinasjoner mellom TU og VAR_c blir vanskelig hvis vi kun ser på adaptive egenskaper basert på TU og entropi. Av den grunn velger vi å også benytte adaptive egenskaper basert på TU og VAR_c .

Vi har tidligere skrevet om at gråtonen er en indikator på hvor stor kromatinansamlingen er på det lokale området, og kan fungere som en prognostisk markør. Siden LBP er gråtoneinvariant kan vi, ved å kombinere TU og senterpikselens gråtone, få med informasjon om gråtonen. I figur 4.11 og 4.12 observerte vi at entropi kombinert med TU ga størst skille mellom prognosegruppene, men vi ser også at TU kombinert med gråtoner gjør det nesten like

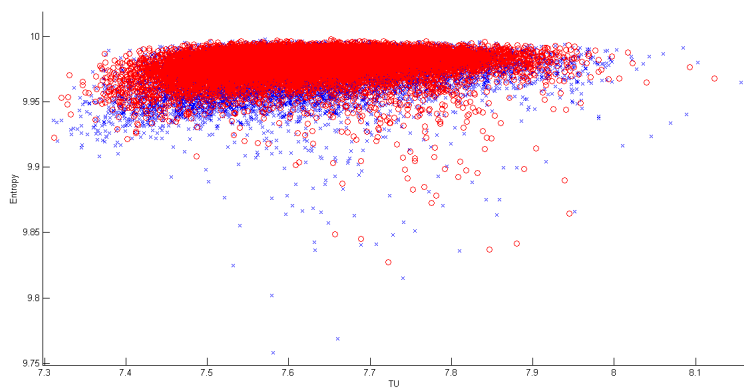
bra. I tillegg ser vi at denne kombinasjonen har ulikheter i andre TU-grupper enn det vi så for entropi og varians. Det vil derfor være av interesse å se nærmere på denne kombinasjonen, ettersom det kun kan være tilfeldigheter som gjør at entropi kombinert med TU tilsynelatende skiller prognosegruppene bedre fra hverandre enn TU kombinert med senterpikselens gråtone. Med andre ord velger vi å utføre alle de tre kombinasjonene.



(a) η som gråtonen til senterpikselen.

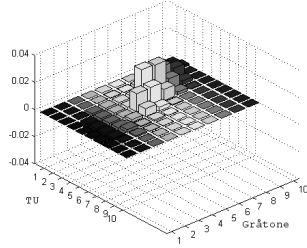


(b) η som $VAR_{c8,1}$.

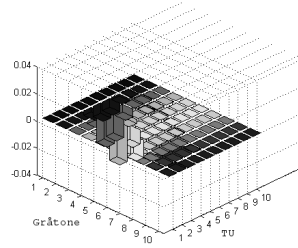


(c) η som $ENT_{8,1}$.

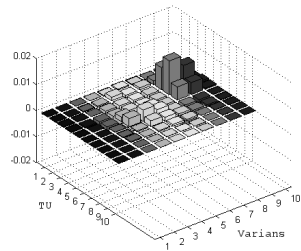
Figur 4.11: Denne figuren viser spredningsplott basert på gjennomsnittlig TU og η -verdi til cellekjerne-bilder, til alle pasientene i treningssettet i M98. Blå X er fra cellekjerne til pasienter med god prognose, mens rød O er fra cellekjerne til pasienter med dårlig prognose.



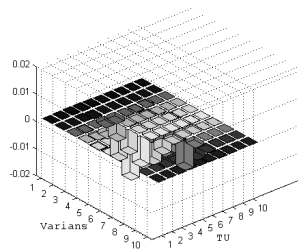
(a) Histogrammet viser differansen (den positive) av andel forekomster av de ulike TU og η , hvor η er gråtonen til senterpikselen. Dataene er hentet fra treningsettet til M98.



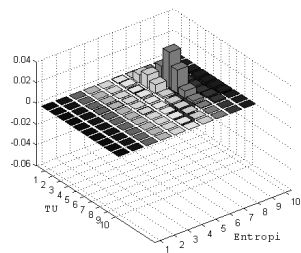
(b) Histogrammet viser differansen (den negative) av andel forekomster av de ulike TU og η , hvor η er gråtonen til senterpikselen. Dataene er hentet fra treningsettet til M98.



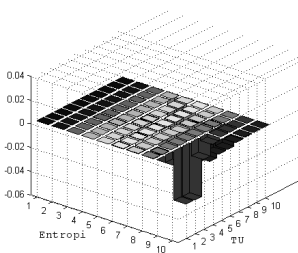
(c) Histogrammet viser differansen (den positive) av andel forekomster av de ulike TU og η , hvor η er $VAR_{c8,1}$. Dataene er hentet fra treningsettet til M98.



(d) Histogrammet viser differansen (den negative) av andel forekomster av de ulike TU og η , hvor η er $VAR_{c8,1}$. Dataene er hentet fra treningsettet til M98.



(e) Histogrammet viser differansen (den positive) av andel forekomster av de ulike TU og η , hvor η er $ENT_{8,1}$. Dataene er hentet fra treningsettet til M98.



(f) Histogrammet viser differansen (den negative) av andel forekomster av de ulike TU og η , hvor η er $ENT_{8,1}$. Dataene er hentet fra treningsettet til M98.

Figur 4.12: Histogrammene viser hvor det er størst differanse i andel forekomster av de ulike TU- og η -gruppene, i forhold til god og dårlig prognose. Differansene er funnet fra normaliserte forekomst-histogrammer, men er ikke normalisert selv.

4.4.5 Bruk av matrisene - Adaptive egenskaper

I seksjon 2.1.1, så vi på bruken av adaptive egenskaper basert på GLCM. Slike adaptive egenskaper kan vi videreføre her, men i stedet for å se på GLCM, ser vi på matrisen som inneholder andelen forekomster av TU-grupper mot returverdiene til $ENT_{8,1}$ og $VAR_{c8,1}$, samt gråtonen til senterpikselen. Med dette finner vi ut hvilke τ og η som oppnår størst klasseskilte mellom prognosegruppene.

Vi starter med å se på to matriser som kombinerer TU-gruppene med η , en for hver prognosegruppe. Vi lar disse matrisene inneholde andelen forekomster av et nabolag som resulterer i en gitt TU-gruppe samt en gitt η -gruppe. Vi kaller disse M_{TH_0} og M_{TH_1} for henholdsvis god og dårlig prognose, og vi tenker på disse som normaliserte. Deretter finner vi avstandene mellom $M_{TH_0}(\tau, \eta)$ og $M_{TH_1}(\tau, \eta)$, for alle τ og η , og setter dem inn i en matrise vi kaller M_D . Vi går frem på lik måte som Nielsen et al. i [45] og benytter disse distansene som vektorer for å finne to egenskaper, F_+ og F_- . På denne måten ender vi opp med to egenskaper som tar hensyn til hvilke forekomster i $M_{TH_{\omega_i}}$ som bidrar til å skille klassene best. Disse egenskapene finnes ved

$$F_+ = \sum_{M_D(i,j) \geq 0} M_{TH_{\omega_i}}(i,j) M_D(i,j)^2 \quad (4.1)$$

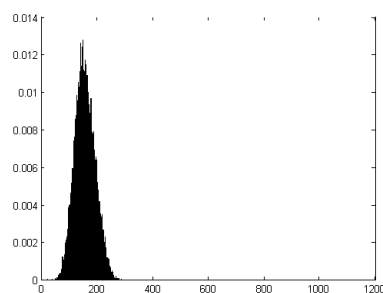
$$F_- = \sum_{M_D(i,j) < 0} M_{TH_{\omega_i}}(i,j) M_D(i,j)^2 \quad (4.2)$$

Før vi kan benytte dette, er vi imidlertid nødt til å finne ut hvilket distansemål som passer best til dataene vi har. Vi har tidligere sett at andelen forekomster av de ulike TU-gruppene har vært tilnærmet normalfordelt. Vi vet derimot ikke hvordan forekomstene blir når vi ser på andelen forekomster av nabolag som resulterer i en gitt TU-gruppe, og en gitt η -gruppe. Av den grunn tar vi for oss noen slike sannsynlighetsfordelinger basert på cellekjerne-bildene i treningssettene i M98 og L23. Et par slike fordelinger er å finne i figur 4.13 og 4.14. Som vi kan se er fordelingene for både god og dårlig prognose omtrent normalfordelte. Dette betyr at vi kan benytte oss av Mahalanobis eller Gaussisk Bhattacharyya distanse. Videre kan vi se at ulikhetene mellom god og dårlig prognose er størst når man ser på variansen. Av den grunn kan det være fordelaktig å bruke Gaussisk Bhattacharyya distanse, slik som vi har gjort tidligere.

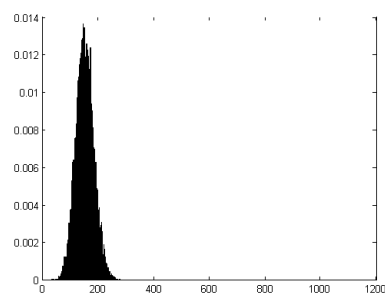
Nå gjenstår det kun å regne ut distansene til alle kombinasjonene av TU- og η -gruppene. Vi husker fra tidligere at Gaussisk Bhattacharyya avstand er gitt ved

$$D_B = \frac{1}{8}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \left(\frac{\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2}{2} \right)^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2} \ln \frac{|\frac{1}{2}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)|}{|\boldsymbol{\Sigma}_1|^{1/2} |\boldsymbol{\Sigma}_2|^{1/2}}$$

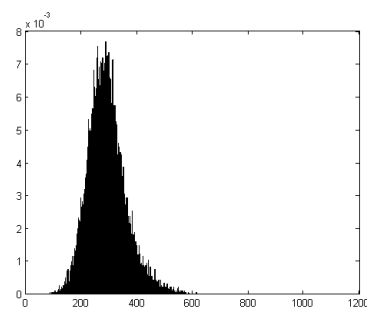
hvor $\boldsymbol{\mu}_i$ og $\boldsymbol{\Sigma}_i$ er henholdsvis forventningsverdi-vektoren og kovariansmatrisen til klasse ω_i . Ettersom vi bare har en egenskap for det vi ønsker å måle



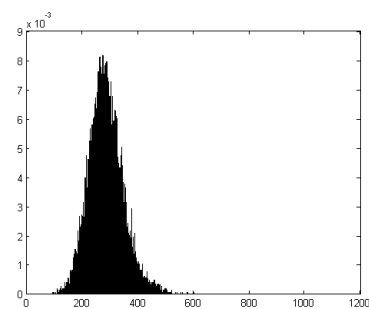
(a) TU-gruppe 6, $ENT_{8,1}$ lik 2.75. God prognose



(b) TU-gruppe 6, $ENT_{8,1}$ lik 2.75. Dårlig prognose

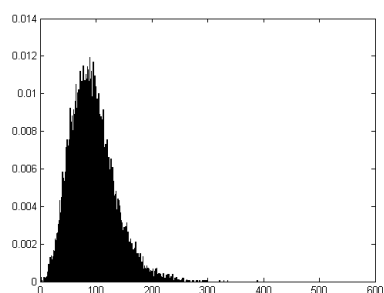


(c) TU-gruppe 4, $ENT_{8,1}$ lik 3. God prognose

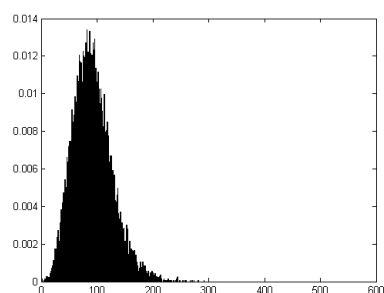


(d) TU-gruppe 4, $ENT_{8,1}$ lik 3. Dårlig prognose

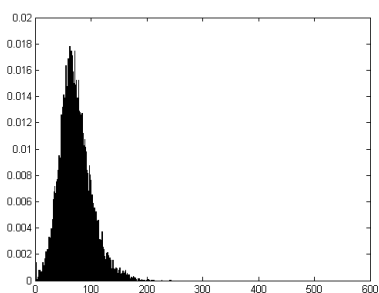
Figur 4.13: Denne figuren viser sannsynlighetsfordelinger til pasienter med god og dårlig prognose fra treningsettet i M98, hvor pasienter med god og dårlig prognose er hver for seg. Sannsynlighetsfordelingene er basert på andelen forekomster av nabolag som gir en TU-gruppe og en entropi-verdi, som er angitt under hver sannsynlighetsfordeling.



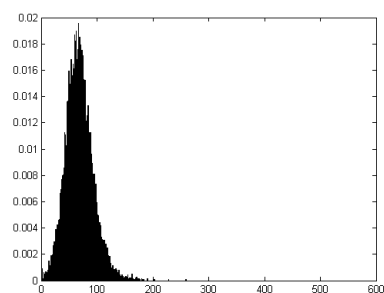
(a) TU-gruppe 4, $VAR_{c8,1}$ -gruppe lik 2.
God prognose



(b) TU-gruppe 4, $VAR_{c8,1}$ -gruppe lik 2.
Dårlig prognose



(c) TU-gruppe 4, $VAR_{c8,1}$ -gruppe lik 3.
God prognose



(d) TU-gruppe 4, $VAR_{c8,1}$ -gruppe lik 3.
Dårlig prognose

Figur 4.14: Denne figuren viser sannsynlighetsfordelinger til pasienter med god og dårlig prognose fra treningsettet i M98, hvor pasienter med god og dårlig prognose er hver for seg. Sannsynlighetsfordelingene er basert på andelen forekomster av nabolag som gir en TU-gruppe og en varians-gruppe, som er angitt under hver sannsynlighetsfordeling.

distanse mellom, og vi ønsker å ha et mål på om distansen er positiv eller negativ, ender vi opp med

$$D_{B1} = \varkappa \left(\frac{1}{4} \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} + \frac{1}{2} \ln \left(\frac{\sigma_1^2 + \sigma_2^2}{2\sigma_1\sigma_2} \right) \right)$$

hvor

$$\varkappa = \begin{cases} 1 & \text{hvis } \mu_1 \geq \mu_2 \\ -1 & \text{ellers} \end{cases}$$

Som vi kan se er vi nødt til å finne forventningsverdien og variansen til hvert element i hver av de to matrisene. Forventningsverdiene kan enkelt finnes ved å normalisere M_{TH_0} og M_{TH_1} , slik at summen av alle elementene i hver av dem blir 1. Av den grunn tenker vi oss fra nå av at $M_{TH_{\omega_i}}$ er normalisert. På den måten vil man finne $\mu_{(\tau,\eta)_{\omega_i}}$ som $M_{TH_{\omega_i}}(\tau, \eta)$, hvor $\mu_{(\tau,\eta)_{\omega_i}}$ representerer forventet andel forekomster av nabolag som gir TU-gruppen τ og entropi-/VAR_c-gruppen/gråtonen η . For å finne variansen, $\sigma_{(\tau,\eta)_{\omega_i}}^2$, til hver forekomst i matrisene setter vi opp en ny matrise, $M_{TH_{\sigma_{\omega_i}^2}}$, for begge prognosegruppene. Et element i en slik matrise vil være variansen til andelen forekomster av nabolag som resulterer i τ og η , basert på alle cellekjernebildene til alle pasientene i en prognosegruppe. D_{B1} blir beregnet for hver τ og η , disse verdiene blir så satt inn i en ny matrise, M_D .

Etter beregning av M_D beregnes F_+ og F_- ut etter formlene (4.1) og (4.2). Disse beregnes for alle cellekjerne-bilder, og sannsynlighetsfordelinger blir satt opp basert på hva egenskapene fra hvert cellekjerne-bilde blir for hver pasient og prognosegruppe. Her er det verdt å merke seg at vi velger å avvike fra fremgangsmåten Nielsen et al. benyttet seg av i [45]. I stedet for å sette opp en sannsynlighetsfordeling, valgte de å la en pasients egenskaper være forventningsverdien, standardavviket og den 10. og 90. prosentandelen i fordelingen. I denne oppgaven velger vi å gå frem på lik måte som vi har gjort i de tidligere seksjonene i dette kapittelet, og på denne måten beholde en viss konsistens mellom fremgangsmåtene.

Etter at sannsynlighetsfordelingene er satt opp finner vi avstanden en pasients sannsynlighetsfordeling har til de to prognosegruppenes sannsynlighetsfordelinger. I figur 4.15 finner vi prognosegruppenes sannsynlighetsfordelinger, for både F_+ og F_- , og disse vurderer vi for å finne ut hvilket distansemål vi skal benytte. Som vi kan se, likner disse ganske mye på sannsynlighetsfordelingene vi så ved for eksempel bruk av $LBP_{8,1}$ på originale cellekjernebilder i L23. Vi kan se at fordelingene kan tilnærmes med en normalfordeling, selv om vi i samtlige fordelinger observerer en liten helling mot den ene siden. I tillegg observerer vi at ulikheten mellom god og dårlig prognose forekommer i standardavviket til disse sannsynlighetsfordelingene, fremfor forventningsverdiene. Av den grunn velger vi å benytte oss av Gaussisk Bhattacharyya, når vi finner avstander mellom pasienters og prognosegruppers sannsynlighetsfordelinger. Dette gjør også at vi opprettholder konsistens mellom de ulike fremgangsmåtene, noe som igjen gjør dem lettere

å sammenlikne i etterkant.

For å undersøke om det er nødvendig å benytte begge de adaptive egenskapene setter vi opp en korrelasjonsmatrise [66] basert på distansene funnet fra sannsynlighetsfordelingne for F_+ og F_- for hver prognosegruppe. Korrelasjonsmatrise basert på treningsettet i M98 blir dermed

	F_+ -god	F_+ -dårlig	F_- -god	F_- -dårlig
F_+ -god	1.0	0.6	1.0	0.6
F_+ -dårlig		1.0	0.6	1.0
F_- -god			1.0	0.6
F_- -dårlig				1.0

Her observerer vi at korrelasjonskoeffisienten mellom F_+ og F_- ble lik 1 (i realiteten < 1 , men > 0.999) i alle tilfeller. Dersom vi nå regner ut et konfidensintervall for korrelasjonen [55], kan vi finne ut med hvor stor sikkerhet de to egenskapene faktisk er korrelert. Vi lar korrelasjonen være $r = 0.999$, og i M98 er størrelsen på treningsettet $N = 20$. Vi starter med å transformere r med Fishers z-transform:

$$z' = \frac{1}{2} \ln\left(\frac{1+r}{1-r}\right) = \frac{1}{2} \ln\left(\frac{1+0.999}{1-0.999}\right) = 3.8$$

For å finne konfidensintervallet benytter vi formelen som generelt er gitt ved

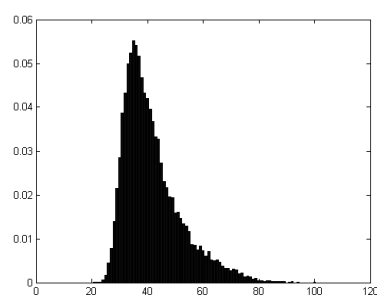
$$z' \pm z\sigma_{z'}$$

hvor z er ønsket konfidensnivå og $\sigma_{z'} = \frac{1}{\sqrt{N-3}}$. Her velger vi å se på et 95% konfidensintervall, noe som gjør at vi får $z = 1.96$. På den måten blir øvre og nedre grense i konfidensintervallet henholdsvis 4.28 og 3.32. For å transformere dette fra z -rommet tilbake til r -rommet benytter vi oss av formelen

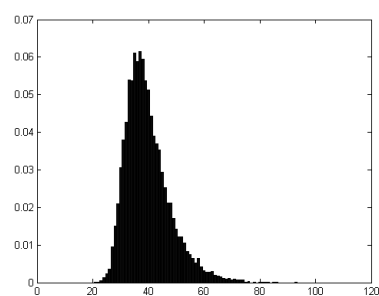
$$r = \frac{e^{2z_b} - 1}{e^{2z_b} + 1}$$

hvor z_b er en av de to z -grensene, avhengig av om man ønsker å finne øvre eller nedre grense i r -rommet. På denne måten ender vi opp med et konfidensintervall på [0.9974 0.9996]. Vi kan med andre ord være veldig sikre på at F_+ og F_- er korrelerte, og av den grunn velger vi å kun bruke F_+ videre.

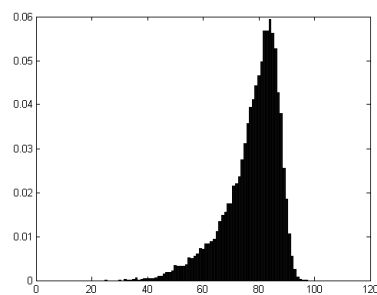
I likhet med tidligere er vi nødt til å velge en klassifikator som bruker distansene til å skille prognosegruppene fra hverandre. Vi har tidligere benyttet Multivariabel Gaussisk klassifikator fordi mange naturlige fenomener kan beskrives på denne måten, og fordi mange distanser tilsynelatende oppfører seg deretter. Det er også verdt å legge merke til at Nielsen et al. også benyttet denne typen klassifikator i [45], hvor de jobbet med adaptive egenskaper. Av disse grunnene, og ønske om å opprettholde konsistens mellom fremgangsmåtene, velger vi å benytte Multivariabel Gaussisk klassifikator. På grunnlag av resultatene fra operatorene benyttet i seksjon 4.1 - 4.3, velger vi å kun anvende de adaptive egenskapene på de originale bildene fra M98 og L23, samt de nedskalerte bildene fra L23. I alle de tre anvendelsene benyttes det operatører med 8 piksels nabolag i de adaptive egenskapene.



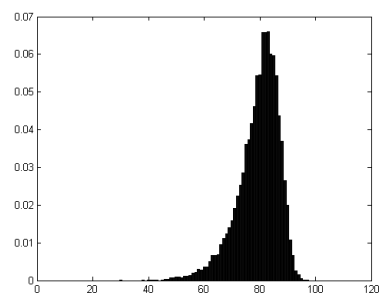
(a) F_+ - God prognose



(b) F_+ - Dårlig prognose



(c) F_- - God prognose



(d) F_- - Dårlig prognose

Figur 4.15: Denne figuren viser sannsynlighetsfordelinger til pasienter med god og dårlig prognose fra treningssettet i M98, hvor pasienter med god og dårlig prognose er hver for seg.

Kapittel 5

Resultater og Diskusjon

I dette kapittelet presenteres resultatene av de forsøkene vi beskrev i kapittel 4. Som vi var inne på i starten av kapittel 4 lar vi sannsynligheten for en klasse $P(\omega_i)$ være lik for begge klassene. Hvis man har et datasett bestående av ulikt antall medlemmer fra hver klasse, vil et klassifiseringsresultat på formen $\frac{\text{Antall riktig klassifisert}}{\text{Antall totalt}}$ bli påvirket av dette antallet. Av den grunn ser vi på forvirringsmatriser ("confusion matrices"), og ut fra disse beregner vi spesifisitet og sensitivitet [67]. Spesifisitet er andelen riktig klassifiserte pasienter med god prognose, mens sensitivitet er andel riktig klassifiserte pasienter med dårlig prognose. For å danne oss et enkelt mål for suksess beregner vi gjennomsnittet mellom disse, og kaller dette for CCR_{eq}.

Som vi skrev om i kapittel 4 vil vi nå se på resultater funnet fra operatører benyttet på 8 og 16 piksels nabolag i originale cellekjerne-bilder fra prostata-materialet M98 og eggstokk-materialet L23, samt operatører benytter på 8 piksels nabolag i nedskalerte cellekjerne-bilder. Disse vil bli tilegnet hver sin seksjon (5.1 - 5.7), Diskusjon som omhandler L23 er å finne i 5.8. Vi vil så gjennomgå resultatene vi oppnådde ved å bruke adaptive egenskaper. Vi starter med resultatene fra M98, før resultatene fra L23 kommer. Etter hver seksjon med resultater kommer det en seksjon med diskusjon av resultatene. Kapittelet blir avsluttet med en mer generell diskusjon omkring bruken av de adaptive egenskapene.

Ideelt sett skal man beskrive alle resultatene og faktaene først, for deretter å ta for seg diskusjonen omkring resultatene. I denne oppgaven er det mange resultater, og for å øke lesbarheten er det en fordel å flette resultater og diskusjon, uten å blande dem. Av den grunn presenteres resultatene fra M98 først, for å bli etterfulgt av en diskusjon omkring disse resultatene. Deretter kommer resultater fra L23, som også blir etterfulgt av en diskusjon. Til slutt kommer resultatene fra bruk av adaptive egenskaper, og diskusjon av disse. Ved å diskutere resultatene fra en gruppe resultater, kan man også lære noe som gjør at man utfører nye beregninger, finner resultater, og så diskuterer disse igjen.

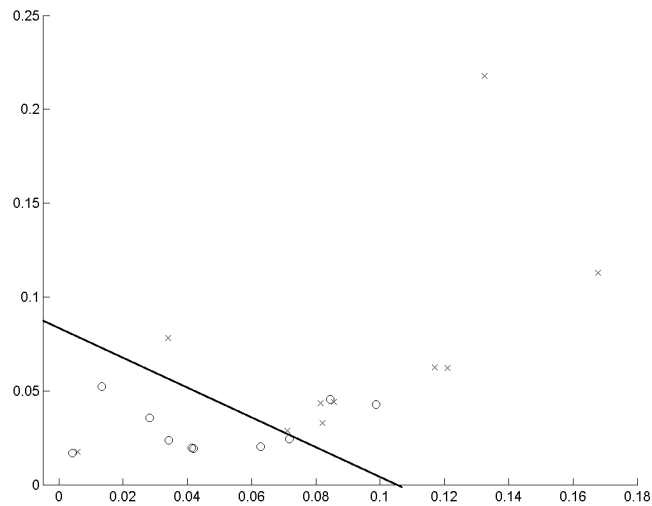
5.1 Resultater fra operatører benyttet på nabolag bestående av 8 piksler i originale cellekjernebilder

Som tittelen tilsier kommer denne seksjonen til å inneholde resultater fra klassifisering med $LBP_{8,1}$ og $VAR_{c8,1}$. I seksjon 4.1.2 skrev vi om at $LBP_{8,1}$ gir ut 17 TU-grupper, og i seksjon 4.1.5 at $VAR_{c8,1}$ gir ut 10 VAR_c -grupper. Det betyr at vi ender opp med 27 resultater fra klassifisering av et treningssett og tilsvarende resultater fra klassifisering av et uavhengig testsett. Treningssettet består av 10 pasienter med god prognose, og 10 pasienter med dårlig prognose. Alle pasienter er fra M98, som er omtalt i kapittel 3.1. Testsettet på sin side består av 8 pasienter med god prognose, og 6 pasienter med dårlig prognose. Før vi benytter klassifikatoren på treningssettet, tar vi for oss spredningsplottet med de 20 pasientene i treningssettet. Det er totalt 27 spredningsplott, hvor x-aksen indikerer en pasients avstand til sannsynlighetsfordelingen til prognosegruppe 0, og y-aksen viser avstanden til sannsynlighetsfordelingen til prognosegruppe 1. Med andre ord blir en pasient vist i plottet basert på pasientens to utregnede Gaussisk Bhattacharyya avstander. Eksempler på slike plott er å finne i figur 5.1. Som vi kan se i spredningsplottene, og som også gjelder for de 15 andre, er at vi kan oppnå forholdsvis gode klassifiseringsresultater. Ved å sette en rett linje mellom klassene i spredningsplottet for TU-gruppe 2, kan vi oppnå en CCReq på 85 %, selv om dette medfører liten margin mellom skillelinjen og noen av pasientene. For TU-gruppe 5 kan vi oppnå en CCReq på 70 %. Uansett ser det ut til at vi kan forvente forholdsvis gode resultater ved bruk av klassifikatoren vi kom frem til i seksjon 4.1.7.

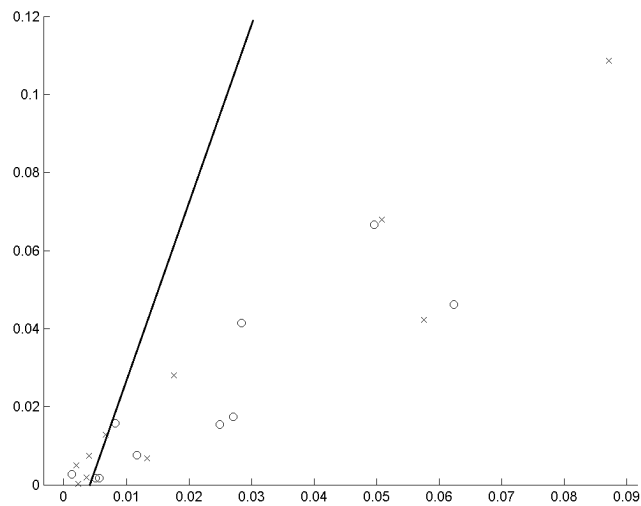
Treningen går ut på å finne en vektor μ og en kovariansmatrise Σ , for hver prognosegruppe, basert på data fra pasientene i treningssettet. μ og Σ blir så brukt i den multivariable gaussiske sannsynlighetsestimatoren, som vi igjen bruker til klassifiseringen av pasienter. Vi lar egenskapene være distansene en pasient har fra sin sannsynlighetsfordeling til de to prognosegruppens sannsynlighetsfordelinger, slik som i spredningsplottene. Med andre ord betyr det at vi ser på klassifiseringer med 2 egenskaper.

5.1.1 Resultater fra bruk av LBP

I tabell 5.1 kan man se hva CCReq fra klassifiseringene basert på de 17 TU-gruppene ble. Ettersom antallet pasienter er likt i begge prognosegruppene, er CCReq lik den totale andelen riktige klassifiseringer. Dersom man ønsker å se nærmere på resultatene, kan man se på tabell 7.1 i Vedlegg 7.2. Her finner man forvirringsmatriser, samt tilhørende total andel riktig klassifiserte pasienter, sensitivitet, spesifisitet og CCReq. Som vi kan se i tabell 5.1 er det TU-gruppe 9 som gjør det best med CCReq på 80 %, i trening, mens TU-gruppe 5 gjør det dårligst med CCReq på 50 %. Gjennomsnittlig CCReq ble 67 %.



(a) Sannsynlighetsfordelingene er basert på TU-gruppe 2. Den rette linjen i spredningsplottet er den skillelinjen som gir høyest CCReq.



(b) Sannsynlighetsfordelingene er basert på TU-gruppe 5.

Figur 5.1: Spredningsplottene viser de 20 pasientene i treningssettet fra M98. Distanse til klassen med god prognose er vist på x -aksen, mens y -aksen viser pasientens distanse til klassen med dårlig prognose. Distansene er funnet ved å beregne Gaussisk Bhattacharyya avstand mellom sannsynlighetsfordelingene.

Tabell 5.1: Denne tabellen viser trenings- og testresultatet for $LBP_{8,1}$ benyttet på de originale bildene. Resultatene er oppgitt som CCReq i %. I tillegg viser tabellen differansen mellom trening og test, i prosentpoeng, samt gjennomsnittlig trening- og testresultat.

TU-gruppe	1	2	3	4	5	6	7	...
Trening	70	75	75	75	50	55	65	...
Test	15	15	15	21	33	25	42	...
Differanse	55	60	60	54	17	30	23	...

...	8	9	10	11	12	13	14	15	16	17	Gj.snitt
...	75	80	70	75	65	60	55	65	70	60	67 %
...	29	21	31	23	15	33	42	25	31	54	28 %
...	46	59	39	52	50	27	13	40	39	6	

Vi går nå over til resultatene ved klassifisering på et eget testsett. I tabell 5.1 kan man se de endelige klassifiseringsresultatene i form av CCReq, og i tabell 7.2 i Vedlegg B kan man som tidligere finne mer utfyllende resultater. Som vi kan se fra resultatene oppnår vi best resultat med TU-gruppe 17, hvor CCReq er 54 %, og dårligst resultat med TU-gruppene 1, 2, 3 og 12 hvor CCReq ble 15 %. Gjennomsnittlig CCReq ble 28 %, noe som vil si et fall på 39 prosentpoeng. Disse resultatene er helt klart ikke hva vi hadde forventet, og i seksjon 5.4.1 diskuterer vi hva som har skjedd, og drøfter hva dette kan komme av.

5.1.2 Resultater fra bruk av varians

Vi går nå gjennom resultatene fra trening og testing med distanser basert på lokal varians, slik som ble beskrevet i seksjon 4.1.4. Resultatene i form av CCReq er å finne i tabell 5.2. Tabell 7.3 i Vedlegg B viser resultatene i mer detalj. Her kan vi se at det er VAR_c -gruppe 4 og 8 som gjør det best på trening, med CCReq på 70 %. VAR_c -gruppe 1 gjør det dårligst med CCReq på 55 %. Gjennomsnittlig CCReq ble 63 %. Dette er litt dårligere resultater enn ved bruk av $LBP_{8,1}$ på treningsettet. Nå gjenstår det å se hvor godt dette generaliseres ved å utføre klassifisering på testsettet.

I tabell 5.2 kan man se at vi, i likhet med bruk av $LBP_{8,1}$, oppnår svært dårlige resultater. Her er det beste resultatet oppnådd av VAR_c -gruppe 4 med CCReq på 44 %, og det dårligste ble oppnådd av VAR_c -gruppe 7 med CCReq på 27 %. Gjennomsnittlig CCReq ble 34 %, et fall på 29 prosentpoeng. Flere detaljer rundt resultatet er å finne i tabell 7.4 i Vedlegg 7.2. Disse resultatene kommer også til å bli et tema i diskusjonen i seksjon 5.4.1.

5.1.3 Kombinering av egenskaper

Til nå har vi sett på resultater fra klassifiseringer hvor 2 egenskaper har blitt brukt om gangen. Til tross for at testresultatene ble dårligere enn vi hadde håpet på, skal vi ta for oss noen kombinasjoner av egenskaper som forhåpentligvis kan gi bedre resultater enn de vi til nå har sett. Ettersom både en TU-gruppe og en VAR_c -gruppe allerede består av 2 egenskaper hver, vil en kombinering av egenskaper føre til at vi minst ender opp med

Tabell 5.2: Denne tabellen viser trening- og testresultatet for $VAR_{c8,1}$ benyttet på de originale bildene. Resultatene er oppgitt som $CCReq$. I tillegg viser tabellen differansen mellom trening og test i prosentpoeng samt gjennomsnittlig trening- og testresultat.

VAR_c -gruppe	1	2	3	4	5	...
Trening	55	60	60	70	60	...
Test	29	31	40	44	40	...
Differanse	26	29	20	26	20	...

...	6	7	8	9	10	Gj.snitt
...	65	60	70	60	65	63 %
...	31	27	29	31	40	34 %
...	34	33	41	29	25	

4 egenskaper. Dette kommer av at det ikke vil være naturlig å splitte opp paret med distanser som kommer fra en TU- eller VAR_c -gruppe. Økning av antall egenskaper kan, som vi har vært inne på tidligere, medføre ”curse of dimensionality”. Dette er vi nødt til å være oppmerksomme på når vi skal gjøre kombineringer her.

Vi starter med å se på 4 egenskaper. Ettersom oppgaven i all hovedsak baserer seg på LBP , vil vi her la kombinasjonen bestå av distansene fra minst en TU-gruppe. Ser vi kun på kombinasjoner mellom en TU-gruppe og en VAR_c -gruppe (som innebærer bruk av 4 egenskaper) vil vi kunne se på $17 \cdot 10 = 170$ ulike kombinasjoner totalt. Ettersom vi benytter LBP i tre ulike fremgangsmåter vil antallet slike kombinasjoner bli $2 \cdot 17 \cdot 10 + 18 \cdot 10 = 520$. Dette er beregnet basert på at $LBP_{8,1}$ kan returnere 17 ulike TU-grupper og benyttes både på originale og nedskalerte cellekjerne-bilder, $LBP_{16,2}$ kan returnere 18 ulike TU-grupper og VAR_c kan returnere 10 ulike VAR_c -grupper. Man kunne også utført kombinasjoner mellom to ulike TU-grupper basert på samme fremgangsmåte, noe som ville resultert i

$$2 \cdot \sum_{i=1}^{16} i + \sum_{i=1}^{17} i = 17 \cdot 16 + \frac{18 \cdot 17}{2} = 425$$

ulike kombinasjoner, når man summerer over alle de tre fremgangsmåtene. For å begrense antall mulige kombinasjoner, kommer vi imidlertid ikke til å se på kombinasjoner mellom ulike TU-grupper funnet med samme fremgangsmåte. I stedet fokuserer vi på kombinasjoner som andre har oppnådd gode resultater med tidligere. Ojala et al. og Ma oppnådde gode resultater ved å kombinere LBP og varians [54] [56] [36], og av den grunn har vi sett på slike kombinasjoner.

I tillegg må vi ta med muligheten til å kombinere TU-grupper som ser på en bestemt andel av cellekjernen med TU-grupper som ser på andre andeler av cellekjernen. Ettersom vi i denne oppgaven begrenser oss til å benytte $LBP_{8,1}$ på originale og nedskalerte cellekjerne-bilder, og $LBP_{16,2}$ på de originale cellekjerne-bildene, vil vi ende opp med $17 \cdot 17 + 17 \cdot 18 + 17 \cdot 18 = 901$ ulike kombinasjoner.

Ojala et al. benyttet kombinasjonen mellom $LBP_{8,1}$ og $LBP_{16,2}$, og endte opp med gode resultater i [54], så dette er en kombinasjon vi ønsker å prøve på våre datasett. Vi har skrevet om bruken av bilde-pyramider, og at man kan hente ut informasjon på ulike nivåer i denne. For å se om en slik representasjon av bildene har noe for seg, ønsker vi også å utføre kombinasjoner av TU fra ulike nivåer i pyramiden. Vi kommer imidlertid ikke til å se på kombinasjoner mellom TU fra $LBP_{8,1}$ benyttet på nedskalerte cellekjerne-bilder og TU fra $LBP_{16,2}$. Disse to operatorene henter informasjon fra omtrent samme deler av cellekjernen, og er mer spennende å sammenlikne med hverandre enn å kombinere. På denne måten begrenser vi antallet ulike kombinasjoner vi kan se på.

Dette fører til $520 + 17 \cdot 17 + 17 \cdot 18 = 1115$ ulike kombinasjoner totalt. Selv om vi har begrenset antallet kombinasjonsmuligheter, er 1115 fortsatt et for stort antall ulike kombinasjoner å se på. Av den grunn velger vi å se på kombinasjoner av TU-grupper og VAR_c -grupper som i all hovedsak gjør det bra i trening.

Det å kombinere de beste enkeltstående egenskapene vil ikke nødvendigvis gi den beste kombinasjonen ettersom korrelasjonen mellom disse ofte er høy [13] [33]. I forsøk på å komme utenom dette kan man bruke en såkalt suboptimal algoritme [33] for å finne en tilnærming til den beste kombinasjonen av egenskaper. Slike algoritmer søker etter den beste kombinasjonen, men resultatet er ikke nødvendigvis optimalt. En slik algoritme er flytende søk, "floating search" [58], som har vist seg å være den beste suboptimale algoritmen [30] [69]. Vi kommer imidlertid ikke til å benytte en suboptimal algoritme for egenskapsutvelgelse. I stedet velger vi å beregne korrelasjonsmatriser for å unngå kombinasjoner hvor egenskapene har for høy korrelasjonskoeffisient. Altså vurderer vi korrelasjonskoeffisienten mellom de tilsynelatende beste egenskapene basert på resultater fra trening og test.

Vi kan huske fra de forrige delseksjonene at det var TU-gruppe 9 og VAR_c -gruppe 4 og 8 som gjorde det best på trening. Men siden VAR_c -gruppe 8 bare oppnådde CCReq på 29 % i test, ser det ikke ut til at den gruppen har så stort potensiale. Dersom man ser på TU-gruppe 17 hadde den 60 % CCReq på trening og 54 % CCReq på test, og var den TU-gruppen med minst fall fra trening til test. Vi kommer derfor til å utføre to klassifiseringer. Den første av disse klassifiseringene baseres på egenskaper fra TU-gruppe 9 kombinert med VAR_c -gruppe 4. Denne kombinasjonen har en korrelasjon på 0.11, se tabell 5.3. Den andre klassifiseringen baseres på kombinasjonen mellom TU-gruppe 17 og VAR_c -gruppe 4 og korrelasjonen er 0.16, se tabell 5.3. Resultatene fra klassifisering utført på testsettet er å finne i tabell 5.4. Som vi kan se ender kombinasjonen av TU-gruppe 17 og VAR_c -gruppe 4 opp med CCReq på 52 %, hvilket er dårligere enn det vi fikk med TU-gruppe 17 alene. Kombinasjonen av TU-gruppe 9 og VAR_c -gruppe 4 oppnådde en CCReq på 21 %. Legg merke til at vi heller ikke her ender opp med noe bedre resultat enn det TU-gruppen oppnådde alene, i tillegg gjorde også VAR_c -gruppe 4 det bedre i test alene.

Tabell 5.3: Korrelasjonsmatrise beregnet fra treningsettet til de originale bildene i M98. Her er det benyttet operatorer med 8 piksels nabolag.

$VAR_c \rightarrow$ $TU \downarrow$	1	2	3	4	5	6	7	8	9	10
1	0.04	0.16	0.15	0.08	0.25	0.01	0.17	0.2	0.2	0.19
2	0.06	0.03	0	0.07	0.39	0.11	0.05	0.08	0.08	0.03
3	0.01	0.11	0.08	0.01	0.29	0.04	0.12	0.15	0.15	0.12
4	0.06	0.01	0.04	0.11	0.39	0.14	0.02	0.05	0.04	0.02
5	0.12	0.26	0.27	0.26	0.21	0.38	0.28	0.28	0.29	0.29
6	0.29	0.37	0.37	0.36	0.25	0.49	0.4	0.39	0.39	0.33
7	0.12	0.13	0.17	0.25	0.51	0.26	0.11	0.09	0.11	0.19
8	0.01	0.02	0.01	0.09	0.34	0.15	0.06	0.08	0.06	0.02
9	0.06	0.19	0.18	0.11	0.19	0.12	0.22	0.24	0.24	0.2
10	0.12	0.1	0.13	0.19	0.44	0.21	0.07	0.04	0.05	0.12
11	0.08	0.04	0.07	0.15	0.41	0.16	0.01	0.02	0	0.06
12	0.02	0.08	0.04	0.05	0.3	0.07	0.11	0.12	0.1	0.03
13	0.25	0.37	0.37	0.34	0.19	0.38	0.38	0.39	0.4	0.36
14	0.18	0.12	0.08	0.02	0.09	0.01	0.16	0.17	0.14	0.04
15	0.11	0.08	0.09	0.15	0.43	0.24	0.06	0.02	0.03	0.08
16	0.14	0.11	0.12	0.18	0.44	0.22	0.09	0.05	0.05	0.09
17	0.23	0.17	0.15	0.16	0.41	0.42	0.22	0.16	0.13	0.1

Tabell 5.4: Testresultater etter kombinerings av egenskaper. Den øverste tabellen viser resultatene etter kombinasjon av TU-gruppe 17 og VAR_c -gruppe 4. Den nederste tabellen viser resultatene etter kombinasjon av TU-gruppe 9 og VAR_c -gruppe 4.

	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	50 %
Gruppe 0:	3	5	8	Spesifisitet:	38 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	5	9	14	CCReq:	52 %
	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	21 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	5	1	6	Sensitivitet:	17 %
Totalt:	7	7	14	CCReq:	21 %

Når vi her kombinerer to egenskaper, er det fire avstander som er med i klassifikasjonen. Som man kan se virker det ikke som om flere egenskaper bidrar til å øke klassifiseringsraten, og det vil derfor være liten grunn til å utføre klassifiseringer med enda flere egenskaper.

5.2 Resultater fra operatorer benyttet på nabolag bestående av 16 piksler i originale cellekjernebilder

I denne seksjonen går vi gjennom resultatene vi fikk fra operatorer som jobber med 16 piksler i de originale cellekjernebildene. Mer konkret tar vi for oss resultatene etter bruk av $LBP_{16,2}$ og $VAR_{c16,2}$, som ble beskrevet i seksjon 4.2. Ettersom fordelingen av utverdier fra $VAR_{c16,2}$ forekommer ulikt i forhold til fordelingen av utverdiene fra $VAR_{c8,1}$, må vi benytte en ny kvantisering. Vi gjør dette slik at de etterfølgende metodene benyttes på VAR_c -grupper som forekommer omtrent like ofte, og på den måten oppnår vi konsistens i den etterfølgende metodebruken. Ved å følge fremgangsmåten

Tabell 5.5: Denne tabellen viser trening- og testresultatet for $LBP_{16,2}$ benyttet på de originale bildene. Resultatene er oppgitt som CCReq i prosent. I tillegg viser tabellen differansen mellom trening og test i prosentpoeng samt gjennomsnittlig trening- og testresultat.

TU-gruppe	1	2	3	4	5	6	7	...
Trening	55	60	60	65	55	55	65	...
Test	63	42	31	42	50	50	42	...
Differanse	-8	18	29	23	5	5	23	...

8	9	10	11	12	13	14	15	16	17	18	Gj.snitt
60	55	65	65	75	70	70	55	55	45	65	60 %
44	33	42	42	33	38	54	50	42	29	23	42 %
16	22	23	23	42	32	20	5	13	16	42	

for kvantisering av varians som ble benyttet i seksjon 4.1.4, ender vi opp med følgende kvantisering:

$$\begin{aligned}
 0 - 4 &\rightarrow 1 \\
 5 - 7 &\rightarrow 2 \\
 8 - 10 &\rightarrow 3 \\
 11 - 14 &\rightarrow 4 \\
 15 - 18 &\rightarrow 5 \\
 19 - 24 &\rightarrow 6 \\
 25 - 32 &\rightarrow 7 \\
 33 - 44 &\rightarrow 8 \\
 45 - 67 &\rightarrow 9 \\
 > 67 &\rightarrow 10
 \end{aligned}$$

5.2.1 Resultater fra bruk av LBP

Tabell 5.5 viser CCReq for alle de 18 TU-gruppene, og et mer utfyllende resultat kan bli funnet i tabell 7.5 i Vedlegg B. Her observerer vi at TU-gruppe 12 gjør det best på treningsettet med en CCReq på 75 %, mens TU-gruppe 17 gjør det dårligst med CCReq på 45 %. Gjennomsnittlig CCReq ble 60 %. Resultatene, i form av CCReq, etter test er å finne i tabell 5.5. Vi observerer at TU-gruppe 1 gjør det best på testsettet med CCReq på 63 %, mens TU-gruppe 18 gjør det dårligst med CCReq på 23 %. Gjennomsnittlig CCReq ble 42 %, noe som gir et fall på 18 prosentpoeng fra trening. Dersom vi tar alle resultatene i betraktning, observerer vi at bruk av 16 piksels nabolag gir bedre resultater enn bruk av $LBP_{8,1}$ på originalbildene. I tillegg observerer vi at resultatene ligger nærme 50 %. Mer utfyllende resultater kan bli funnet i tabell 7.6 i Vedlegg B.

5.2.2 Resultater fra bruk av varians

Tabell 5.6 viser resultatene etter klassifisering av treningsettet i form av CCReq, mens tabell 7.7 i Vedlegg B viser mer utfyllende resultater. Her

Tabell 5.6: Denne tabellen viser trening- og testresultatet for $VAR_{c16,2}$ benyttet på de originale bildene. Resultatene er oppgitt som CCR_{eq} i prosent. I tillegg viser tabellen differansen mellom trening og test i prosentpoeng samt gjennomsnittlig trening- og testresultat.

VAR_c -gruppe	1	2	3	4	5	...
Trening	55	55	55	55	70	...
Test	40	29	19	33	33	...
Differanse	15	26	36	22	37	...

...	6	7	8	9	10	Gj.snitt
...	55	50	55	55	65	57 %
...	83	44	21	25	40	37 %
...	-28	6	34	30	25	

observerer vi at VAR_c -gruppe 5 gjør det best med CCR_{eq} på 70 %, mens VAR_c -gruppe 7 gjør det dårligst med CCR_{eq} på 50 %. Gjennomsnittlig CCR_{eq} ble 57 %. Resultatene etter klassifiseringen av testsettet er å finne i tabell 5.6 i form av CCR_{eq}, mens et mer detaljert resultat er vist i tabell 7.8 i Vedlegg B. Fra disse tabellene observerer vi at det beste resultatet på testsettet blir oppnådd med VAR_c -gruppe 6 med CCR_{eq} på 83 %, mens dårligst ut kommer VAR_c -gruppe 3 med CCR_{eq} på 19 %. Gjennomsnittlig CCR_{eq} ble 37 %, noe som er et fall på 20 prosentpoeng fra trening. I tillegg observerer vi at kun VAR_c -gruppe 6 oppnår en CCR_{eq} over 50 %, og at flere resultater er dårligere enn de vi så for $VAR_{c8,1}$ benyttet på originalbildene. Til tross for mange dårlige resultater ved bruk av $VAR_{c16,2}$ på testsettet går vi nå videre til å se på muligheten for kombinerings av egenskaper.

5.2.3 Kombinering av egenskaper for klassifisering

I seksjon 5.1.3 tok vi for oss resultater etter kombinasjoner mellom TU- og VAR_c -grupper, men vi skrev også at kombinasjoner mellom TU-grupper funnet fra ulik bruk av LBP også var noe vi kom til å se nærmere på. Som vi har vært inne på tidligere kom Ojala et al. [54] frem til at kombinerings av $LBP_{P,R}$ med ulikt nabolag og radius ga bedre resultater, enn ved å bruke kun en type $LBP_{P,R}$. I stedet for å regne ut distanser med en ny LBP-operator som ser på de samme pikslene som både $LBP_{8,1}$ og $LBP_{16,2}$, jobber vi heller med kombinasjoner mellom en TU-gruppe fra $LBP_{8,1}$ og en TU-gruppe fra $LBP_{16,2}$. Dette vil ikke nødvendigvis gi et like godt bilde av hvordan teksten i det lokale området oppfører seg, ettersom både $LBP_{8,1}$ og $LBP_{16,2}$ oppnår rotasjonsinvariante mønstre uavhengig av hverandre, men vi unngår imidlertid en operator som baserer TU-gruppene på binære mønstre av lengde 24.

Før vi tar for oss kombinasjon mellom TU-grupper fra $LBP_{8,1}$ og $LBP_{16,2}$, starter vi med kombinasjoner mellom TU og VAR_c -grupper funnet med de to 16-nabolags-baserte operatorene. Korrelasjonen mellom TU- og VAR_c -gruppene er satt opp i tabell 5.7. Vi husker fra tidligere i denne seksjonen at den TU-gruppen som gjorde det best i klassifiseringen av treningsettet var TU-gruppe 12. Vi så også at TU-gruppe 14 kom rett bak med CCR_{eq} på 70

%. Dersom vi tar hensyn til testresultatene ser vi at TU-gruppe 14 gjorde det en del bedre, og av den grunn kan det være en fordel å velge TU-gruppe 14 fremfor 12. Dersom vi ser tilbake på resultatene fra bruk av VAR_c , ser vi at VAR_c -gruppe 5 ga best resultat ved klassifisering av treningsettet. I tabell 5.7 observerer vi at VAR_c -gruppen er svært lite korrelert til de to TU-gruppene, og ettersom korrelasjonskoeffisienten er 0.07 i begge tilfellene velger vi å kombinere VAR_c -gruppe 5 med TU-gruppe 14. Klassifiseringsresultatet etter test, med denne kombinasjonen, er å finne i tabell 5.8. Her observerer vi at CCR_{eq} ender på 46 %.

Etter test basert på VAR_c observerte vi at VAR_c -gruppe 6 var den eneste gruppen som oppnådde resultater bedre enn 50 %, og av den grunn kunne det vært fristende å se på en kombinasjon med denne gruppen. Hvis vi derimot ser på korrelasjonen mellom de to aktuelle TU-gruppene og denne VAR_c -gruppen, kan vi se at den laveste korrelasjonskoeffisienten er 0.2, hvilket er mer enn dobbelt så høyt som kombinasjonen vi allerede har valgt oss ut. I tillegg til dette var ikke resultatet etter trening særlig godt for VAR_c -gruppe 6, hvilket tyder på at testresultatet kommer av tilfeldigheter. Av disse grunnene kommer vi ikke til å jobbe med kombinasjoner med denne gruppen.

Den andre kombinasjonen blir, som beskrevet tidligere, mellom TU-grupper utregnet fra $LBP_{8,1}$ og $LBP_{16,2}$. Vi har tidligere kommet frem til at TU-gruppe 9 var en av de beste kandidatene fra $LBP_{8,1}$, og i denne seksjonen har vi allerede vært inne på at TU-gruppe 14 er en god kandidat for $LBP_{16,2}$. Korrelasjonskoeffisienten mellom TU-gruppe 9 fra $LBP_{8,1}$ og TU-gruppe 14 fra $LBP_{16,2}$ beregnes til 0.29. I 5.5 så vi at TU-gruppe 12 og 13 gjorde det henholdsvis bedre og like bra i trening som TU-gruppe 14. Korrelasjonskoeffisienten mellom TU-gruppe 9 fra $LBP_{8,1}$ og TU-gruppe 12 og 13 fra $LBP_{16,2}$ er beregnet til henholdsvis 0.24 og 0.03. Med andre ord er korrelasjonskoeffisienten mellom TU-gruppe 9 fra $LBP_{8,1}$ og TU-gruppe 13 fra $LBP_{16,2}$ mye lavere enn i de to andre kombinasjonene. Av den grunn velger vi å utføre kombinasjonen mellom TU-gruppe 9 fra $LBP_{8,1}$ og TU-gruppe 13 fra $LBP_{16,2}$, og på denne måten ender vi opp med testresultatet vist i tabell 5.9. Her observerer vi at resultatet ikke ble særlig godt med en CCR_{eq} på 35 %.

Tabell 5.7: Korrelasjonsmatrise beregnet fra treningsettet til de originale bildene i M98. Her er det benyttet operatører med 16 piksels nabolag.

$VAR_c \rightarrow$ $TU \downarrow$	1	2	3	4	5	6	7	8	9	10
1	0.04	0.02	0.05	0.07	0.02	0.4	0.06	0.07	0.1	0.13
2	0.15	0.14	0.16	0.16	0.1	0.45	0.07	0.17	0.19	0.2
3	0.39	0.45	0.41	0.38	0.21	0.26	0.48	0.45	0.4	0.31
4	0.17	0.15	0.16	0.17	0.09	0.48	0.07	0.18	0.2	0.21
5	0.05	0.04	0.07	0.1	0.09	0.36	0.02	0.09	0.12	0.15
6	0.03	0.02	0.01	0.05	0.08	0.31	0.1	0.03	0.07	0.12
7	0.18	0.17	0.12	0.06	0.01	0.16	0.2	0.11	0.05	0.02
8	0.21	0.24	0.19	0.12	0.11	0.29	0.31	0.21	0.16	0.04
9	0.25	0.25	0.25	0.23	0.04	0.55	0.14	0.24	0.27	0.24
10	0.04	0.07	0.11	0.12	0.04	0.47	0.03	0.09	0.15	0.17
11	0.07	0.09	0.12	0.13	0.03	0.5	0.02	0.12	0.16	0.16
12	0.22	0.21	0.17	0.14	0.07	0.44	0.11	0.18	0.16	0.17
13	0.12	0.27	0.28	0.3	0.23	0.05	0.23	0.24	0.24	0.21
14	0.07	0.07	0.03	0.01	0.07	0.2	0.12	0.02	0.03	0.09
15	0.03	0.02	0.02	0.06	0.13	0.22	0.07	0.03	0.07	0.12
16	0.05	0.03	0.04	0.07	0.12	0.35	0.07	0.05	0.08	0.13
17	0.24	0.27	0.24	0.23	0.17	0.03	0.36	0.29	0.23	0.11
18	0.2	0.33	0.32	0.33	0.21	0.14	0.37	0.35	0.33	0.27

Tabell 5.8: Resultater etter kombinasjon av TU-gruppe 14 og VAR_c -gruppe 5. TU og VAR_c er funnet ved operatører som ser på 16 piksels nabolag i originalbildene. Klassifiseringen er gjort på testsettet.

	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	4	10	14	CCReq:	46 %

Tabell 5.9: Resultater etter kombinasjon av TU-gruppe 9 fra $LBP_{8,1}$ og TU-gruppe 13 fra $LBP_{16,2}$. Begge operatorene er benyttet på originalbildene. Klassifiseringen er utført på testsettet.

	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	3	5	8	Spesifisitet:	38 %
Gruppe 1:	4	2	6	Sensitivitet:	33 %
Totalt:	7	7	14	CCReq:	35 %

5.3 Resultater fra operatører benyttet på nabolag bestående av 8 piksler i nedskalerte cellekjerne-bilder

Vi har vært inne på to ulike måter for å se på forskjellige andeler av cellekjernen, og i denne seksjonen tar vi for oss resultatene man får fra klassifisering basert på operatører med 8 piksels nabolag benyttet på nedskalerte cellekjerne-bilder. Når vi her skriver om nedskalerte bilder har vi i denne oppgaven halvert de to aksene, noe som medfører at det totale antall piksler i bildet blir $\frac{1}{4}$ av hva det var i det opprinnelige bildet. Resultatene er oppnådd ved å følge fremgangsmåten i seksjon 4.3. Ved kvantisering av varians har vi benyttet fremgangsmåten i seksjon 4.1.5, hvilket resulterte i følgende kvantisering:

0 – 6	→ 1
7 – 10	→ 2
11 – 14	→ 3
15 – 19	→ 4
20 – 24	→ 5
25 – 31	→ 6
32 – 41	→ 7
42 – 57	→ 8
58 – 86	→ 9
> 86	→ 10

5.3.1 Resultater fra bruk av LBP

Vi starter med resultatene fra klassifisering av treningsettet, og man kan finne en oversikt over hva CCR_{eq} ble for trening i tabell 5.10. Mer utfyllende resultater er å finne i tabell 7.9 i Vedlegg B. I disse tabellene observerer vi at TU-gruppe 6 og 15 gjør det best med CCR_{eq} på 70 %, mens TU-gruppe 2-5, 8 og 10-12 gjør det dårligst med CCR_{eq} på 55 %, når man ser bort fra TU-gruppe 17. Grunnen til at vi ser bort fra TU-gruppe 17 er at det finnes for få forekomster av TU som tilhører denne gruppen til at vi får dannet sannsynlighetsfordelinger til de ulike pasientene. Gjennomsnittlig CCR_{eq} ble 56 %.

Testresultatene er å finne i tabell 5.10, og mer detaljerte resultater er å finne i tabell 7.10 i Vedlegg B. Her observerer vi at det, i likhet med resultatene for treningsettet, er TU-gruppe 6 som gjør det best med CCR_{eq} på 69 %. TU-gruppe 15, som også gjorde det best i trening, er TU-gruppen som ender opp med svakest resultat ved testing med CCR_{eq} på 40 %. Gjennomsnittlig CCR_{eq} ble 53 %, noe som tilsvarer et fall på 3 prosentpoeng fra trening. Generelt kan vi se at resultatene etter test med nedskalerte bilder

Tabell 5.10: Denne tabellen viser trening- og testresultatet for $LBP_{8,1}$ benyttet på de nedskalerte bildene. Resultatene er oppgitt som CCReq i prosent. I tillegg viser tabellen differansen mellom trening og test i prosentpoeng samt gjennomsnittlig trening- og testresultat.

TU-gruppe	1	2	3	4	5	6	7	...
Trening	65	55	55	55	55	70	60	...
Test	63	50	63	63	63	69	50	...
Differanse	2	5	-8	-8	-8	1	10	...

...	8	9	10	11	12	13	14	15	16	17	Gj.snitt
...	55	65	55	55	55	65	65	70	60	0	56 %
...	50	63	50	50	63	54	60	40	42	0	53 %
...	5	2	5	5	-8	11	5	30	18	0	

Tabell 5.11: Denne tabellen viser treningsresultatet for VAR_c benyttet på de nedskalerte bildene. Resultatene er oppgitt som CCReq i prosent. I tillegg viser tabellen differansen mellom trening og test i prosentpoeng samt gjennomsnittlig trening- og testresultat.

VAR_c -gruppe	1	2	3	4	5	...
Trening	50	45	70	65	55	...
Test	40	38	33	60	58	...
Differanse	10	7	37	5	-3	...

...	6	7	8	9	10	Gj.snitt
...	55	60	50	60	65	57 %
...	52	38	29	25	54	43 %
...	3	22	21	35	11	

gir bedre resultater, enn ved bruk av originalbildene. Vi kan også se at testresultatene er i mer samsvar med resultatene fra trening enn det vi så for originalbildene.

5.3.2 Resultater fra bruk av varians

I tabell 5.11 kan man finne resultatene i form av CCReq etter klassifisering av treningssettet. Her observerer vi at VAR_c -gruppe 3 gjør det best med CCReq på 70 %, mens VAR_c -gruppe 2 er den som gjør det dårligst med CCReq på 45 %. Gjennomsnittlig CCReq ble 57 %. Flere detaljer rundt de ulike resultatene kan bli funnet i tabell 7.11 i Vedlegg B. I tabell 5.11 observerer vi at resultatene etter testing ikke er like gode som etter trening. Her er det VAR_c -gruppe 4 som gjør det best med CCReq på 60 %, mens VAR_c -gruppe 9 gjør det dårligst med CCReq på 25 %. Gjennomsnittlig CCReq ble 43 %, hvilket er et fall på 14 prosentpoeng fra trening. I tillegg kan vi se at 6 av 10 resultater er lavere enn 50 %, og at resultatene generelt sett er trukket mot 0 % fremfor 100 %. Det kan derfor se ut til, i motsetning til for TU, at VAR_c fortsatt sliter med tilsvarende problemer som ved bruk av originalbildene. Vi kan imidlertid se at testresultatene har blitt litt bedre ved bruk av nedskalering, og at det med andre ord ser ut til at benyttelse av VAR_c på en større andel av cellekjernen, i likhet med $LBP_{8,1}$, har noe for seg. Resultatene drøftes i seksjon 5.4.3.

5.3.3 Kombinering av egenskaper

I denne seksjonen tar vi for oss kombinasjoner av TU og VAR_c -grupper, samt kombinasjoner av en TU-gruppe hentet fra de nedskalerte bildene og en hentet fra de originale bildene. På denne måten får vi sett hvor godt bruken av flere nivåer i en bildepyramide gjør det. Vi benytter også her korrelasjonsmatriser.

Vi starter med å se på kombinasjon av en TU-gruppe og en VAR_c -gruppe. I likhet med tidligere baserer vi valget av TU- og VAR_c -gruppe hovedsakelig på resultatene vi fikk i treningsettet. Vi så tidligere i denne seksjonen at TU-gruppe 6 og 15 gjorde det best på trening, men kun gruppe 6 klarte å videreføre de gode resultatene til testing. Av den grunn er TU-gruppe 6 den beste kandidaten av disse. Blant VAR_c -gruppene var det gruppe 6 og 10 som gjorde det best. Vi kan se i tabell 5.12 at korrelasjonen mellom TU-gruppe 6 og VAR_c -gruppe 6 er 0.52, mens korrelasjonen mellom TU-gruppe 6 og VAR_c -gruppe 10 er 0.01. Av den grunn velger vi å kombinere TU-gruppe 6 og VAR_c -gruppe 10.

Resultatene etter denne kombineringsen er å finne i tabell 5.13. Som man kan se endte resultatet av denne kombineringsen opp med en CCR_{eq} på 54 %, noe som ikke er bedre enn det vi fikk med TU-gruppe 6 alene. I tabell 5.14 kan man se resultatene fra tilsvarende kombineringsen for originalbildet. Her ble resultatene langt dårligere, noe som bidrar til å støtte opp om bruken av nedskalerte cellekjerne-bilder fremfor de originale når man ser på cellekjernene i M98 datasettet.

Ved kombineringsen av distanser fra en TU-gruppe i de to ulike skaleringsene av cellekjerne-bildene gjelder det å velge en TU-gruppe som ser ut til å kunne gjøre det greit i begge skaleringsene. I tabell 5.1 og 5.10 observerer vi at TU-gruppe 9 kommer relativt godt ut av det i begge skaleringsene for trening. I tillegg er korrelasjonen mellom de to TU-gruppene lav, med en korrelasjonskoeffisient på 0.04. Vi kommer derfor til å utføre klassifiseringen basert på disse to. Resultatet fra klassifiseringen gjort på testsettet er å finne i tabell 5.15. Som vi kan se ender denne kombinasjonen opp med en CCR_{eq} på 52 %, og man kan se at resultatet havner mellom CCR_{eq} oppnådd med TU-gruppe 9 i hver av de to skaleringsene.

Tabell 5.12: Korrelasjonsmatrise beregnet fra treningsettet til de nedskalerte bildene i M98. Her er det benyttet operatører med 8 piksels nabolag.

$VAR_c \rightarrow$ $TU \downarrow$	1	2	3	4	5	6	7	8	9	10
1	0.24	0.13	0.19	0.45	0.59	0.41	0.23	0.12	0.08	0.02
2	0.16	0.04	0.08	0.37	0.56	0.34	0.15	0.05	0	0.09
3	0.13	0.04	0.13	0.48	0.63	0.36	0.15	0.04	0	0.06
4	0.23	0.1	0.12	0.33	0.55	0.39	0.21	0.11	0.05	0.06
5	0.1	0.03	0.14	0.58	0.68	0.35	0.13	0.02	0.01	0.02
6	0.34	0.26	0.3	0.44	0.59	0.52	0.36	0.24	0.17	0.01
7	0.07	0.02	0.02	0.27	0.41	0.2	0.05	0	0	0.03
8	0.13	0.02	0.03	0.25	0.51	0.31	0.12	0.04	0	0.07
9	0.35	0.21	0.25	0.47	0.56	0.47	0.32	0.19	0.11	0.03
10	0.1	0.01	0.02	0.29	0.52	0.29	0.09	0.01	0.04	0.12
11	0.1	0.01	0	0.26	0.52	0.29	0.09	0.01	0.04	0.1
12	0.1	0.01	0.06	0.36	0.6	0.34	0.12	0.02	0.03	0.12
13	0.22	0.12	0.2	0.48	0.59	0.4	0.22	0.12	0.07	0.03
14	0.2	0.13	0.18	0.39	0.57	0.42	0.24	0.13	0.07	0.1
15	0.18	0.21	0.15	0.28	0.4	0	0.19	0.22	0.23	0.09
16	0.09	0	0.02	0.19	0.5	0.29	0.09	0.03	0.02	0.08
17	-	-	-	-	-	-	-	-	-	-

Tabell 5.13: Resultater etter kombinerings av TU-gruppe 6 og VAR_c -gruppe 10. Teksturanalysen er gjort på nedskalerte cellekjerne-bilder i testsettet.

	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	50 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	1	5	6	Sensitivitet:	83 %
Totalt:	3	11	14	CCReq:	54 %

Tabell 5.14: Resultater etter kombinerings av TU-gruppe 6 og VAR_c -gruppe 10. Teksturanalysen er utført på de originale cellekjerne-bildene i testsettet.

	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	3	11	14	CCReq:	40 %

Tabell 5.15: Resultater etter kombinerings av TU-gruppe 9 fra originale og nedskalerte cellekjerne-bilder i testsettet.

	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	50 %
Gruppe 0:	3	5	8	Spesifisitet:	38 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	5	9	14	CCReq:	52 %

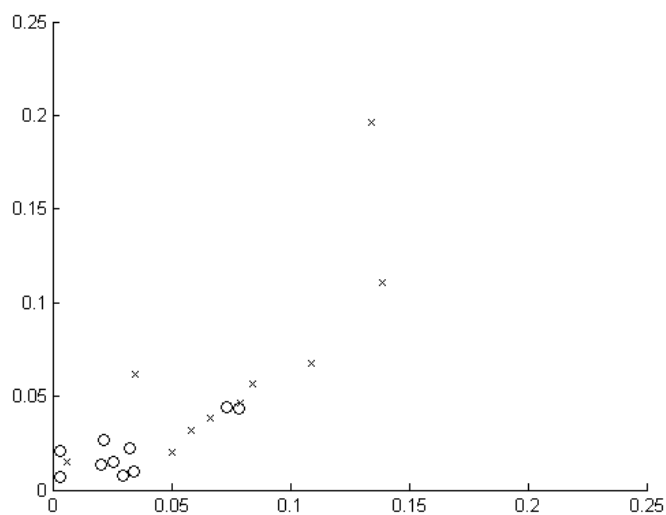
5.4 Diskusjon rundt datasettet M98

Fra resultatene i M98 observerte vi at klassifikatoren i flere tilfeller klarte å oppnå gode resultater på treningsettet. Dette ga oss en indikasjon på at det var noe som utgjorde en forskjell mellom klassene. Da det kom til test observerte vi at klassifiseringsresultatene i flere tilfeller ble langt fra resultatene som vi så i trening. Dette gjaldt spesielt resultatene fra operatører benyttet på 8 piksels nabolag i originalbildene. Selv om det samme forekom for operatører basert på 16 piksels nabolag, var det ikke like tydelig her. Som vi skrev innledningsvis tar vi for oss hver av de tre fremgangsmåtene før en mer generell diskusjon til slutt.

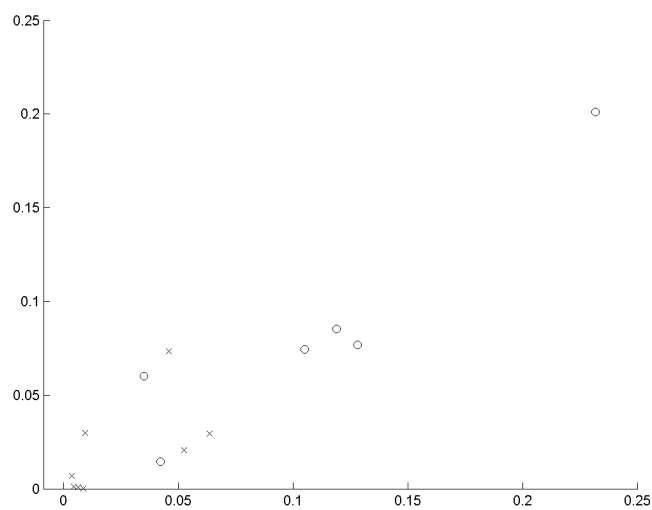
5.4.1 Operatører på 8 piksels nabolag i originalbildene

Fra resultatene i seksjon 5.1 husker vi at klassifiseringsraten ble langt dårligere etter test, i forhold til det vi endte opp med i trening. Vi husker for eksempel at CCR_{eq} fra TU-gruppe 3 etter trening var på 75 %, men etter testing falt denne verdien til 15 %. Ved å se på spredningsplottene for de to tilfellene kan vi få et enda bedre bilde av hva som skjer. Disse er å finne i figur 5.2.

Her observerer vi at flertallet av pasientene med god prognose i treningsettet har høyere distanser enn flertallet av pasienter med dårlig prognose i samme datasett. I spredningsplottet til pasientene i testsettet, observerer vi derimot at det er pasientene med dårlig prognose som har de høye distansene, mens pasientene med god prognose har distanser nærmere 0. Dette er også tilfellet for de andre TU-gruppene. Vi ser med andre ord tydelig hvorfor klassifiseringsresultatene blir som de blir. Spørsmålet som gjenstår er hvorfor distansene innad i en pasientgruppe endres så ekstremt fra trening til test, spesielt med tanke på at distansene blir målt til samme sannsynlighetsfordeling i de to klassifiseringene. Dersom vi ser på noen av diagnosegruppens sannsynlighetsfordelinger, observerer vi at flere sannsynlighetsfordelinger forandrer seg forholdsvis mye for en prognosegruppe i de to settene. Prognosegruppens sannsynlighetsfordelinger i de to datasettene til TU-gruppe 3 er å finne i figur 5.3. Det kan se ut til at pasientene i trening- og testsettet er forholdsvis ulike hverandre når man kun ser på pasientene i en prognosegruppe. Dette medfører at vi tilsynelatende har for høy variasjon mellom pasientene i en prognosegruppe i forhold til antall pasienter som finnes i trening- og testsettet, og som gjør at vi ikke ender opp med en tilstrekkelig generalisering av resultatene. Denne variasjonen, det lave antallet pasienter, samt litt uflaks i inndelingen av test- og treningsettet er tilsynelatende grunnen til at vi endte opp med såpass snudde resultater i test sammenliknet med trening. For å bli helt sikre på at det er dette som er tilfellet, er vi imidlertid nødt til å utføre tilsvarende forsøk på et uavhengig datasett, gjerne med flere pasienter. Dette får vi blant annet gjort når vi senere analyserer datasettet L23, hvor det totale antallet pasienter er 134. Vi vil komme tilbake til dette i seksjon 5.8. Vi ser også nærmere på bruken av større datasett i seksjon 5.4.5.

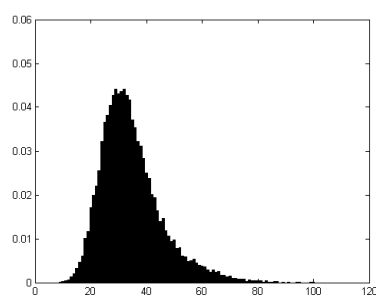


(a) Pasienter fra treningssettet

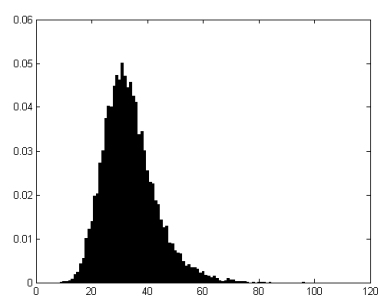


(b) Pasienter fra testsettet

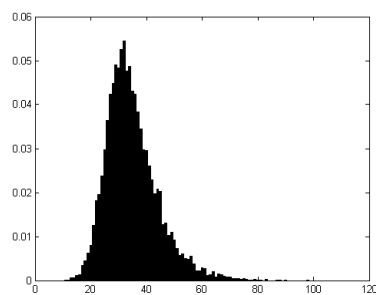
Figur 5.2: Denne figuren viser spredningsplott hvor en pasient er plottet i forhold til distansene som er utregnet basert på sannsynlighetsfordelingene til TU-gruppe 3. Spredningsplottene inneholder pasientene fra trening- og testsettet hver for seg. X representerer pasienter med god prognose, mens O representerer pasienter med dårlig prognose.



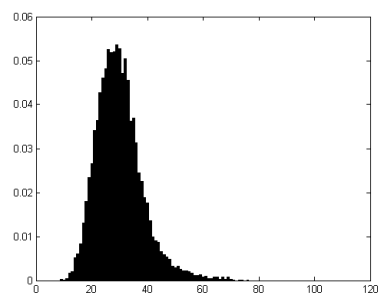
(a) God prognose i treningsettet.
 $\mu = 34.6, \sigma^2 = 122.2$



(b) Dårlig prognose i treningsettet.
 $\mu = 33.6, \sigma^2 = 123.2$



(c) God prognose i testsettet.
 $\mu = 34.8, \sigma^2 = 127.8$



(d) Dårlig prognose i testsettet.
 $\mu = 29.9, \sigma^2 = 149.5$

Figur 5.3: Figuren viser sannsynlighetsfordelingene til TU-gruppe 3 basert på alle pasientene i en prognosegruppe i ett av de to datasettene. Dette er vist for begge prognosegruppene og både trening- og testsett.

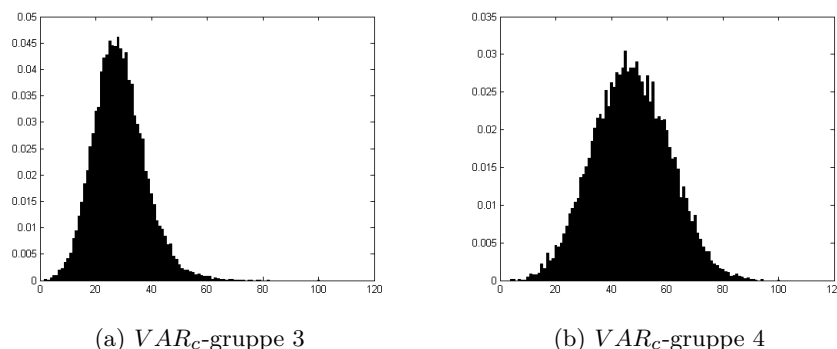
5.4.2 Operatører benyttet på 16 piksels nabolag i originalbildene

Når vi nå går over til resultatene fra operatører basert på 16 piksels nabolag, observerer vi at resultatene er litt bedre enn de vi oppnådde ved bruk av operatorene som baserte seg på 8 piksels nabolag i de originale cellekjernebildene. Vi tar først for oss resultatene fra bruken av TU-gruppene, og observerer i flere tilfeller greie resultater etter trening, men at de faller igjennom i test. Med andre ord observerer vi den samme tendensen vi så tidligere fra bruken av $LBP_{8,1}$ på originalbildene.

Hvis vi nå går videre til resultatene etter bruk av $VAR_{c16,2}$ observerer vi først og fremst at heller ikke disse resultatene er spesielt gode. I motsetning til resultatene fra de to tilfellene hvor $VAR_{c8,1}$ er brukt, observerer vi at $VAR_{c16,2}$ ikke en gang oppnår gode resultater på treningsettet. Dette tyder derfor på at det er lite som skiller pasientene med god og dårlig prognose når vi ser på distansene mellom sannsynlighetsfordelinger basert på $VAR_{c16,2}$. Dette trenger ikke nødvendigvis være på grunn av at det ikke finnes noe skille mellom prognosegruppene når man benytter $VAR_{c16,2}$. Lave klassifiseringsresultater avhenger også av distansemåleren og klassifikatoren som er brukt, som begge har sine antagelser om dataen vi ser på. Disse ble valgt på grunnlag av hvordan $VAR_{c8,1}$ oppførte seg på originalbildene. Vi kan imidlertid observere at sannsynlighetsfordelingene basert på $VAR_{c16,2}$ i de fleste tilfeller er tilnærmet normalfordelte. Dette kan man se i figur 5.4, hvor sannsynlighetsfordelingene til VAR_c -gruppe 3 og 4 er vist. Dette støtter med andre ord opp om bruk av en distansemåler som antar normalfordeling, slik som vi allerede benytter. I resultatene fra klassifiseringen av testsettet kan vi observere at nesten alle CCReq er en del lavere enn 50 %, og enkelte er lavere enn 30 %. Hvis det ikke hadde eksistert en forskjell mellom god og dårlig prognose, kunne man forventet at samtlige CCReq skulle ligge nærmere 50 %. Dette tyder på at det faktisk kan finnes et skille mellom god og dårlig prognose, men at dette skillet ikke var tydelig nok i treningsettet. Det ser nok en gang ut til at vi har et tilfelle hvor for få pasienter i trening og test medfører usikkerhet rundt resultatene, så man skal være forsiktig med å trekke konklusjoner basert på disse resultatene.

5.4.3 Operatører benyttet på 8 piksels nabolag i de nedskalerte bildene

Da vi så på resultatene fra de nedskalerte cellekjernebildene merket vi oss at resultatene var bedre enn ved bruk av originalbildene. Dette gjaldt spesielt for resultatene vi fikk ved bruk av $LBP_{8,1}$. Ved bruk av $LBP_{8,1}$ benyttet på originale cellekjerne-bilder observerte vi at gjennomsnittlig CCReq kom på 28 % etter test, mens bruk av $LBP_{8,1}$ på nedskalerte cellekjerne-bilder ga gjennomsnittlig CCReq på 56 %. Hvis man derimot beregner gjennomsnittlig CCReq ved å se bort ifra TU-gruppe 17, ender man opp med 60 %. Ser man på resultatene fra de nedskalerte bildene i trening og testing, observerer man at disse er forholdsvis like. Ettersom flere av testresultatene



Figur 5.4: Sannsynlighetsfordelinger basert på to ulike $VAR_{c16,2}$ -grupper

er høyere enn 50 %, ser det ut til at denne fremgangsmåten har noe for seg ved separasjon av prognosegruppene. Som vi husker var det TU-gruppe 6 som kom best ut av det i test, og denne ser ut til å utgjøre et godt verktøy når man skal skille prognosegruppene fra hverandre i generelle tilfeller.

Ettersom resultatene i trening og test er forholdsvis like, ser det ut til at variasjonen mellom sannsynlighetsfordelingene til pasientene i en prognosegruppe er lavere enn de vi ender opp med etter bruk av $LBP_{8,1}$ på originalbildene. I figur 5.5 kan man se spredningsplottene basert på TU-gruppe 3 for både trening og test. Her observerer vi at det er mye mer samsvar mellom distansene til pasientene i en prognosegruppe i trening og test, i forhold til det vi så fra tilsvarende TU-gruppe i originalbildene.

I figur 5.6 kan vi se sannsynlighetsfordelingene til de to prognosegruppene, for både trening og test. Her kan vi se at de to sannsynlighetsfordelingene til dårlig prognose er veldig like hverandre. Dette viser at variasjonen mellom de ulike pasientene med dårlig prognose er relativt lav, og at vi med andre ord har en god modell for disse pasientene. Vi kan imidlertid se at sannsynlighetsfordelingene til prognosegruppen som representerer god prognose er litt mer ulike hverandre. Det kan derfor se ut til at vi kunne oppnådd bedre resultater hvis vi hadde hatt flere pasienter til å få disse tilnærmingene til sannsynlighetsfordelingene mer like. På samme tid er sannsynlighetsfordelingene til god prognose forholdsvis ulike sannsynlighetsfordelingene til prognosegruppen som representerer dårlig prognose. Av den grunn er det fortsatt rom for greie klassifiseringsresultater. Dette støtter antakelsen om at bruken av nedskalerte cellekjerne-bilder fører til mindre variasjon i sannsynlighetsfordelingene, basert på TU, når man ser på pasientene i en prognosegruppe.

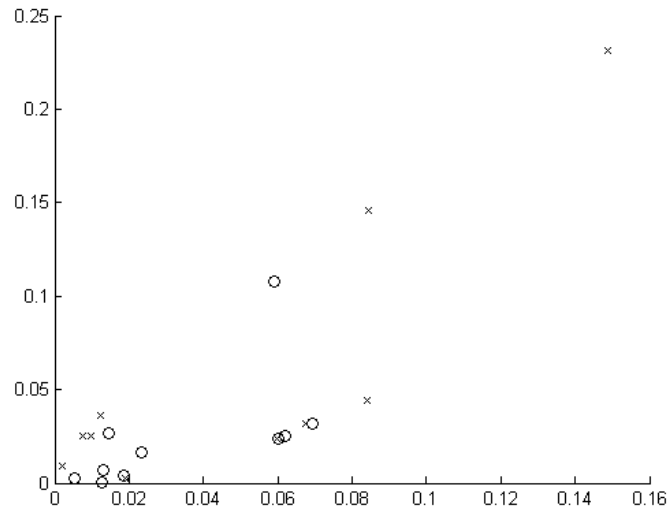
Resultatene etter bruk av $VAR_{c8,1}$ på de nedskalerte cellekjerne-bildene ble ikke like gode som resultatene etter bruk av $LBP_{8,1}$ på de samme bildene. Her lå flertallet av CCR_{eq} under 50 % etter test, og i de tilfellene hvor CCR_{eq} er høyere enn 50 % er CCR_{eq} ikke mye over 50 %. Gjennomsnittlig CCR_{eq} kom på 43 %, hvilket er høyere enn gjennomsnittet for CCR_{eq}

basert på $VAR_{c,1}$ på originalbildene som ble 34 %. Med andre ord ser vi at nedskalering av cellekjerne-bildene medfører litt bedre resultater når man måler den lokale variansen, men i likhet med tidligere har vi fortsatt et problem med generalisering fra trening til test. Det kan derfor tenkes at også denne operatoren har behov for å bli benyttet på flere pasienter før klassifisering, slik vi så for operatorene i seksjon 5.4.1.

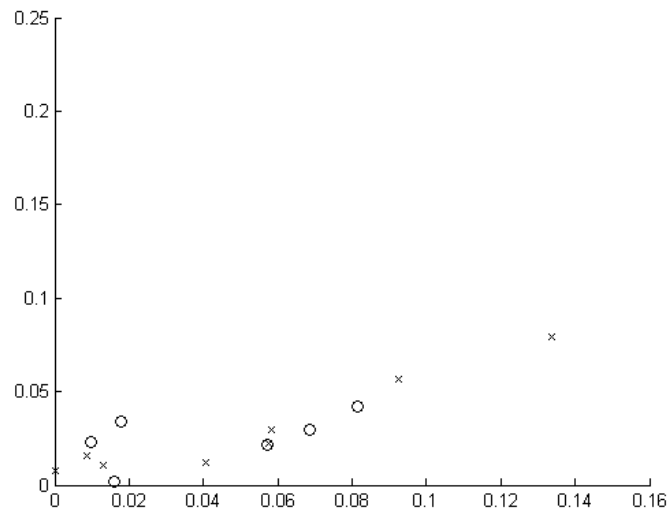
5.4.4 Kombinering av egenskaper

Ved kombinerings av egenskaper har vi observert at bruken av 4 egenskaper ikke medfører bedre resultater enn det vi oppnådde med 2. I enkelte tilfeller ble resultatene mye dårligere enn det 2 av de 4 egenskapene klarte å gi alene. Det er derfor grunn til å tro at vi er et offer for "curse of dimensionality", noe som lett kan forekomme når vi ser på små datasett, slik vi gjør her. I et forsøk på å komme utenom dette problemet kunne vi ha prøvd å slå sammen distansene fra samme $TU-/VAR_c$ -gruppe ved hjelp av for eksempel "Principal Component Analysis" (PCA) [40], eller "Fishers Linear Discriminant" [46]. I flere av spredningsplottene basert på de to distansene, fra en TU - eller VAR_c -gruppe, kan vi se at det å tilordne det 2-dimensjonale tilfellet ned til en dimensjon er mulig, uten å miste for mye klaseskillende informasjon. Dette kan man blant annet se i spredningsplottet for distansene fra VAR_c -gruppe 1 som ble funnet med $VAR_{c,1}$ på de originale cellekjerne-bildene. Spredningsplottet er å finne i figur 5.7. Dette gjelder imidlertid ikke for alle plottene, og i disse kan man risikere å miste klaseskillende informasjon hvis man slår distansene sammen. Med andre ord ser sammenslåing av distanser lovende ut for noen $TU-/VAR_c$ -grupper, og mindre lovende ut for andre. Til tross for at man risikerer tap av informasjon ved en slik sammenslåing, kan det tenkes at sammenslåtte egenskaper kan gi bedre resultater når de kombineres med hverandre enn det vi har oppnådd tidligere.

Vi har vært inne på at det i noen spredningsplott er slik at pasientene tilnærmet havner på en linje, mens det i andre spredningsplott ikke er tilfellet. I spredningsplottene hvor pasientene ikke forekommer på en linje, kan man i noen tilfeller finne en parabel med symmetri-akse $d_0 = d_1$, hvor d_0 og d_1 er distansene til henholdsvis god og dårlig prognose. Dette kan man blant annet se i figur 5.8a som viser spredningsplottet basert på TU -gruppe 5 fra $LBP_{8,1}$ benyttet på originalbildene. Hvis vi hadde latt distansene få tilegnet et fortegn basert på forventningsverdien til en pasients sannsynlighetsfordeling, i forhold til forventningsverdien til prognosegruppen man finner avstanden til, kan man i mange tilfeller gå fra 2 linjer i plottet til 1. I figur 5.8b kan vi observere hva som skjer med spredningsplottet fra figur 5.8a hvis man gir distansene fortegn. Som vi kan se blir det nå lettere å gå fra det 2-dimensjonale tilfellet ned til én dimensjon. Vi må imidlertid være oppmerksomme på at vi ikke lenger kan skille pasientene like godt som tidligere. For å oppnå bedre resultater kunne vi ha laget en ny egenskap som baserte seg på den absolutte avstanden fra origo, og på den måten endt opp med en tilnærming til hvordan pasientene forekom tidligere. Dette ville imidlertid ført til at det å inkludere fortegn av distansene hadde blitt me-

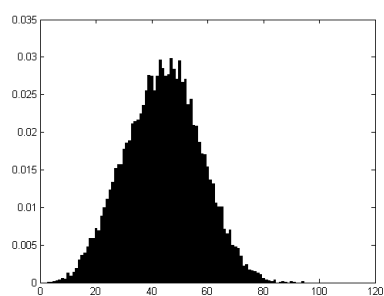


(a) Pasienter fra treningsettet

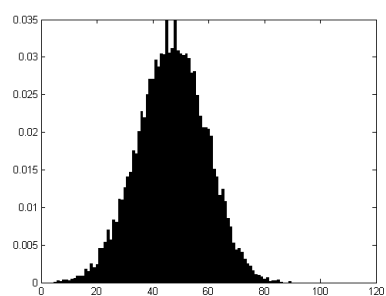


(b) Pasienter fra testsettet

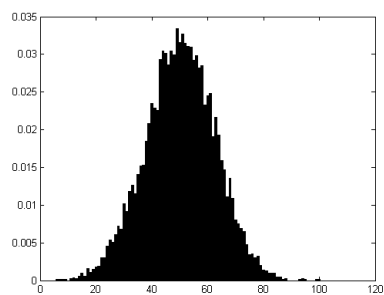
Figur 5.5: Figuren viser spredningsplott hvor en pasient er plottet i forhold til distansene som er utregnet basert på sannsynlighetsfordelingene til TU-gruppe 3. Spredningsplottene inneholder pasientene fra trening- og testsettet hver for seg. Cellekjerne-bildene er nedskalert før $LBP_{8,1}$ er benyttet. X representerer pasienter med god prognose, mens O representerer pasienter med dårlig prognose.



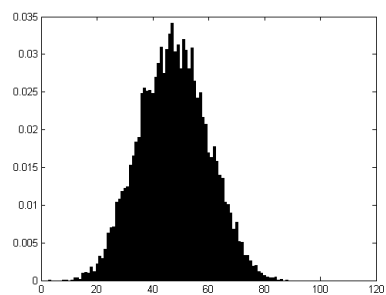
(a) God prognose i treningsettet



(b) Dårlig prognose i treningsettet

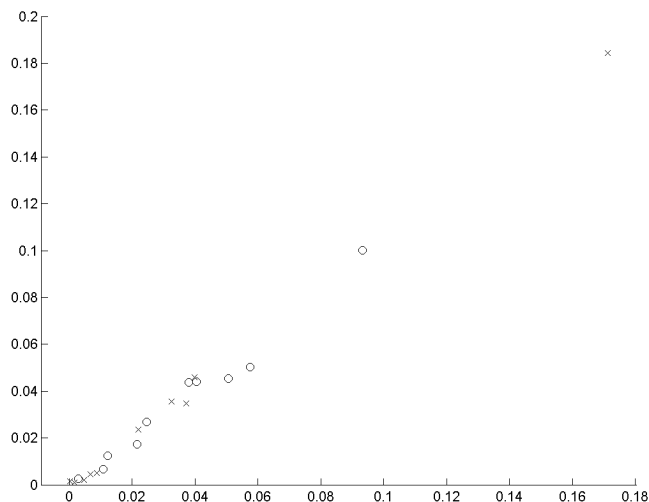


(c) God prognose i testsettet



(d) Dårlig prognose i testsettet

Figur 5.6: Figuren viser sannsynlighetsfordelingene til TU-gruppe 3, til alle pasientene i en prognosegruppe. Dette er vist for begge prognosegruppene i både trening og testsettet til M98, og er basert på nedskalerte cellekjerne-bilder.



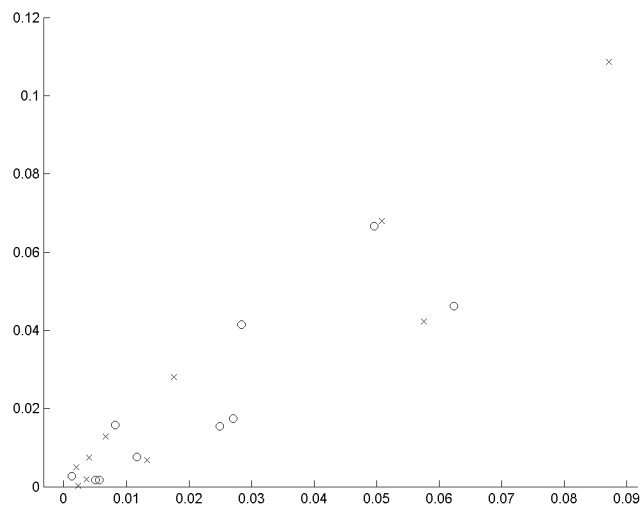
Figur 5.7: Figuren viser spredningsplott hvor en pasient er plottet i forhold til distansene som er utregnet basert på sannsynlighetsfordelingene til VAR_c -gruppe 1. Spredningsplottene inneholder pasientene fra treningsettet.

ningsløst. I tillegg ville målingen av distansen fra et punkt i plottet til origo, som vi gjorde om til en egenskap, blitt tilnærmet lik den egenskapen vi ville endt opp med ved bruk av PCA. Denne fremgangsmåten ville med andre ord ikke fått oss lenger enn det vi kom til i forrige avsnitt.

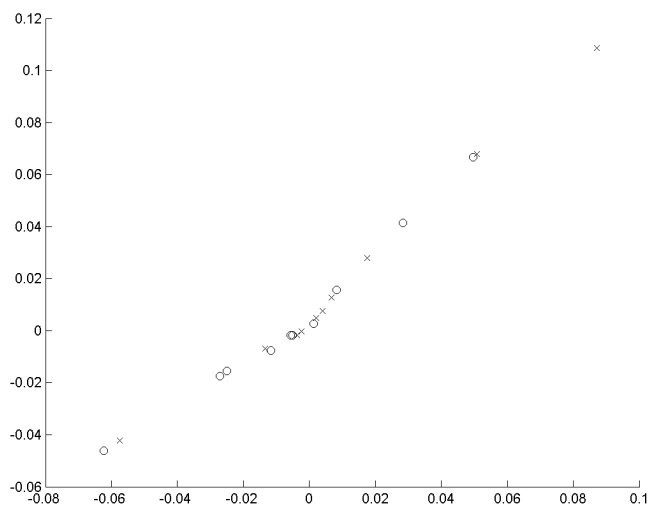
5.4.5 Litt mer generell diskusjon rundt M98

Hvis vi nå går vekk fra å se på kombineringer, og heller tar for oss resultatene fra M98 som en helhet, kan vi trekke konklusjonen om at $LBP_{8,1}$ benyttet på cellekjerne-bilder med nedskalerte akser gjør det best. Det er også det eneste tilfellet hvor trening- og testresultater ser ut til å stemme overens med hverandre. Dårligste resultater får vi ved bruk operatorer basert på 8 piksels nabolag benyttet på originalbildene. Dette tyder på at teksturanalyse i form av LBP og VAR_c benyttet på en litt større andel av bildet, er veien å gå. Dette gjelder spesielt for LBP . Her er det viktig å huske på at mikroskopoppløsningen er 162 nm per piksel i originalbildet, og at vi analyserer cellekjerne i prostatakrefte. Dette trenger ikke å være tilfellet i andre former for kreft, eller for bilder med en annen oppløsning. Som påpekt tidligere er det også et lite datasett vi tar for oss her. Av den grunn kan vi ikke si med sikkerhet hva som fungerer og ikke i generelle tilfeller.

Når vi tar alle resultatene oppnådd ved bruk av VAR_c i betraktning, observerer vi at ingen av de tre ulike anvendelsene gir særlig gode resultater. I likhet med for LBP er det $VAR_{c8,1}$ benyttet på originalbildene som gjør det dårligst, mens $VAR_{c8,1}$ benyttet på nedskalerte bilder gjør det best. Dette støtter dermed påstanden om at bruk av VAR_c på en større andel av cellekjerne-bildet medfører bedre resultater. Vi har også sett at hverken



(a) Spredningsplott basert på distansene til TU-gruppe 5, hvor $LBP_{8,1}$ er benyttet på originalbildene.



(b) Lik som (a), men her tar distansene hensyn til negative verdier.

Figur 5.8: Denne figuren viser spredningsplott hvor en pasient fra M98 er plottet i forhold til distansene som er utregnet basert på sannsynlighetsfordelinger til TU-gruppe 5. X representerer pasienter med god prognose, mens O representerer pasienter med dårlig prognose.

kombinering av $LBP_{P,R}$ og $VAR_{cP,R}$ eller LBP -operatorer med ulike skaleringer av enten bildene operatorene benyttes på, eller antallet piksler i nabolaget operatoren ser på om gangen, ga bedre resultater enn det vi fikk i flere tilfeller med $LBP_{P,R}$ alene. Nok en gang må vi imidlertid understreke at det lave antallet pasienter er med på å trekke flere resultater ned.

Som vi var inne på i kapittel 4 er en pasient med på å påvirke sannsynlighetsfordelingen til prognosegruppen pasienten er med i, når man måler avstandene til denne pasienten. Ettersom vi har såpass få pasienter vil denne påvirkningen kunne ha en del å si. Vi vet at halvparten av egenskapene til pasientene er påvirket av dette, og ettersom disse er med på å avgjøre forventningsverdiene og kovariansmatrisen, som igjen blir brukt i klassifiseringen av testsettet, vil dette kunne være med å påvirke klassifiseringsresultatene. Vi har imidlertid sett at sannsynlighetsfordelingene med alle pasientene er ganske ulike hverandre i trening og test, for samme prognosegruppe, med unntak av $LBP_{8,1}$ benyttet på nedskalerte cellekjerne-bilder. Vi kunne ikke håpet på å oppnå mer likhet mellom sannsynlighetsfordelingene om vi hadde tatt bort en pasient, når denne pasientens avstand skulle måles. I stedet er det ikke urimelig å tenke at sannsynlighetsfordelingene i mange tilfeller kunne blitt mer ulike hverandre, når man ser på sannsynlighetsfordelingene i en prognosegruppe. Det betyr at vi ikke kunne forventet veldig mye bedre resultater etter test, selv om vi hadde tatt ut en pasient fra prognosegruppens sannsynlighetsfordeling, når denne pasientens distanser skulle bli målt.

For å validere om de dårlige resultatene vi har oppnådd skyldes bruk av for få pasienter i datasettet, fikk vi testet $LBP_{8,1}$ og $VAR_{c8,1}$ på en større del av M98, ikke bare supertreningsett 1 og 2. Dette settet består av 89 pasienter hvor 49 pasienter har god prognose, og 40 har dårlig. Inndelingen i trening- og testsett er gjort ved å velge annenhver pasient til de to datasettene. Denne valideringen gjelder kun de originale bildene, og ble utført av Tarjei Sveinsgjerd Hveem som er doktorgradstudent ved Radiumhospitalet. Ved testing av $LBP_{8,1}$ ble 62 % CCReq det beste resultatet, mens 46 % CCReq ble det dårligste resultatet. Gjennomsnittet av CCReq på test ble 59 %. For $VAR_{c8,1}$ ble det beste resultatet 59 %, og det dårligste 43 %. Gjennomsnittet av CCReq på test ble 52 %. Disse resultatene viser at de ekstremt dårlige resultatene vi oppnådde på test skyldtes for få pasienter i datasettet.

5.5 Resultater fra bruk av operatorer benyttet på nabolag bestående av 8 piksler på originale bilder fra L23

I seksjon 5.1 fikk vi veldig uventede resultater på testing, og vi erfarte at dette kom av for få pasienter i M98. L23 har 67 pasienter til trening og 67 pasienter til testing, og tanken er at vi vil kunne komme oss unna problemet med bruk av for få pasienter. Vi har utført tilsvarende forsøk på L23 som vi gjorde på M98. Dette vil fortelle oss om den samme fremgangsmåten vil ha noe for seg på eggstokk-materiale, og samtidig vil det økte antallet pasienter gi et bedre grunnlag for mer generelle konklusjoner. Ved siden av dette er L23 et datasett det er utført mange ulike forsøk på tidligere, hvilket innebærer gode sammenlikningsgrunnlag for resultatene. Vi vil først se på resultatene for bruk av $LBP_{8,1}$, for deretter å se på resultatene for bruk av VAR_c . Her er det brukt lik kvantisering som i seksjon 4.1.5.

5.5.1 Resultater fra bruk av LBP

I tabell 5.16 finner man en oversikt over CCReq for alle TU-grupper, og mer utfyllende resultater er å finne i tabell 7.13 i Vedlegg C. Fra tabellen observerer vi at TU-gruppe 17 oppnår best resultat i trening, med en CCReq på 75 %. Det dårligste resultatet oppnås av TU-gruppe 5 med CCReq på 48 %. Gjennomsnittlig CCReq ble 59 %. Når vi går over til testing er resultatene å finne i tabell 5.16, samt mer detaljert i tabell 7.14 i Vedlegg C. Her observerer vi at TU-gruppe 14 gjør det best med CCReq på 63 %. Den TU-gruppen som gjør det dårligst er gruppe 5, med CCReq på 41 %. Gjennomsnittlig CCReq ble 50 %, noe som er et fall på 9 prosentpoeng fra trening. Her observerer vi at testen ikke faller slik vi så for M98, og vi vil komme tilbake til dette i seksjon 5.8.1.

5.5.2 Resultater fra bruk av varians

Resultatene etter bruk av VAR_c på treningsettet er å finne i tabell 5.17. Her observerer man at VAR_c -gruppe 4 gjør det best, med CCReq på 63 %, mens de som gjør det dårligst er gruppe 6, 8 og 10 med CCReq på 52 %.

Tabell 5.16: Denne tabellen viser trening- og testresultatet for $LBP_{8,1}$ benyttet på de originale bildene i L23. Resultatene er oppgitt som CCReq. I tillegg viser tabellen differansen mellom trening og test i prosentpoeng samt gjennomsnittlig trening- og testresultat.

TU-gruppe	1	2	3	4	5	6	7	...
Trening	61	51	59	58	48	62	53	...
Test	49	47	47	45	41	61	52	...
Differanse	12	4	12	13	7	1	1	...

...	8	9	10	11	12	13	14	15	16	17	Gj.snitt
...	63	53	66	65	53	57	59	58	60	75	59 %
...	46	50	50	48	44	55	63	48	53	55	50 %
...	17	3	16	17	9	2	-4	10	7	20	

Tabell 5.17: Denne tabellen viser trening- og testresultatet for VAR_c benyttet på de originale bildene i L23. Resultatene er oppgitt som CCReq. I tillegg viser tabellen differansen mellom trening og test i prosentpoeng samt gjennomsnittlig trening- og testresultat.

VAR_c -gruppe	1	2	3	4	5	...
Trening	58	54	56	63	59	...
Test	40	58	56	56	46	...
Differanse	18	-4	0	7	13	...

...	6	7	8	9	10	Gj.snitt
...	52	53	52	56	52	56 %
...	52	56	57	61	52	53 %
...	0	-3	-5	-5	0	

Gjennomsnittlig CCReq ble 56 %. For mer utfyllende resultater kan man se på tabell 7.15 i Vedlegg C. Når det kom til test, var det VAR_c -gruppe 9 som gjorde det best med CCReq på 61 %. Dette kan man se i tabell 5.17. Dårligst resultat på testing oppnådde VAR_c -gruppe 1 med CCReq på 40 %. Gjennomsnittlig CCReq ble 53 %, noe som er et fall på kun 3 prosentpoeng fra trening. For flere detaljer rundt resultatet kan man se på tabell 7.16 i Vedlegg C. I likhet med bruk av $LBP_{8,1}$ på originale cellekjerne-bilder snur ikke resultatene seg her heller, slik vi så etter bruk av samme fremgangsmåter i M98. Mer diskusjon omkring dette kommer i seksjon 5.8.1.

5.5.3 Kombinering av egenskaper

Fra trening observerer vi at TU-gruppe 17 gjør det bra i trening, men faller igjennom i test. TU-gruppe 6 oppnår ikke et like godt resultat i trening som TU-gruppe 17, men gjør det fortsatt bedre enn de fleste TU-gruppene. Ved testing forblir resultatet omtrent uendret med et fall på 1 prosentpoeng fra 62 % til 61 % CCReq, se tabell 5.16. Dette gjør at TU-gruppe 6 er en god kandidat blant TU-gruppene til kombinering. I tabell 5.17 observerer vi at VAR_c -gruppe 4 gjør det best på trening, og faller kun med 7 prosentpoeng til test. Av den grunn velger vi ut denne VAR_c -gruppen. Det er imidlertid verdt å legge merke til at flere av VAR_c -gruppene som gjør det bra i trening faller litt igjennom i test. Den beste VAR_c -gruppen som ikke gjør dette, er gruppe 3. Ettersom denne gruppens resultater ikke blir dårligere i test, velger vi å se på denne også. Fra tabell 5.18 kan vi se at en kombinasjon av TU-gruppe 6 og VAR_c -gruppe 4 har en korrelasjon på 0.03, mens en kombinasjon av TU-gruppe 6 og VAR_c -gruppe 3 har en korrelasjon på 0.01. Resultatene etter kombineringene er å finne i tabell 5.19.

Vi husker fra resultatene ved bruk av $LBP_{8,1}$ og $VAR_{c8,1}$ på originalbildene i M98 at kombinasjonen vi endte opp med var mellom TU-gruppe 9 og VAR_c -gruppe 4. Vi endte også opp med å bruke VAR_c -gruppe 4 i kombinasjonen i forrige avsnitt, men vi valgte også å bruke TU-gruppe 6, og ikke 9. Det kunne imidlertid vært interessant å se om kombinasjonen vi benyttet i M98 kan ha noe for seg, og av den grunn ønsker vi å utføre en kombinasjon mellom TU-gruppe 9 og VAR_c -gruppe 4 basert på dataene i L23. I tabell

Tabell 5.18: Korrelasjonsmatrise beregnet fra treningsettet til de originale bildene i L23. Her er det benyttet operatorer med 8 piksels nabolag.

$VAR_c \rightarrow$ $TU \downarrow$	1	2	3	4	5	6	7	8	9	10
1	0.04	0.21	0.39	0.53	0.58	0.13	0.18	0.22	0.22	0.13
2	0.04	0.33	0.53	0.65	0.63	0.19	0.3	0.36	0.35	0.23
3	0.06	0.2	0.39	0.54	0.61	0.11	0.16	0.22	0.23	0.15
4	0.01	0.3	0.51	0.65	0.66	0.14	0.25	0.33	0.34	0.24
5	0.11	0.02	0.01	0	0.21	0.23	0.06	0.04	0	0.03
6	0.04	0.01	0.01	0.03	0.04	0.07	0.03	0.04	0.04	0.02
7	0.18	0.07	0.32	0.52	0.56	0.01	0.05	0.14	0.2	0.14
8	0.01	0.3	0.51	0.68	0.71	0.1	0.25	0.33	0.35	0.28
9	0.03	0.22	0.43	0.58	0.66	0.23	0.21	0.28	0.29	0.17
10	0.03	0.27	0.47	0.61	0.62	0.12	0.23	0.29	0.31	0.22
11	0.01	0.28	0.47	0.61	0.62	0.12	0.23	0.29	0.3	0.2
12	0.12	0.18	0.43	0.59	0.61	0.04	0.13	0.22	0.26	0.19
13	0.03	0.22	0.43	0.56	0.57	0.11	0.18	0.27	0.32	0.26
14	0.02	0.01	0.02	0.06	0.04	0.06	0.04	0	0.06	0.19
15	0.02	0.28	0.44	0.56	0.58	0.14	0.25	0.28	0.27	0.19
16	0.05	0.29	0.41	0.51	0.52	0.13	0.25	0.26	0.23	0.15
17	0.14	0.13	0.07	0.02	0.01	0.1	0.12	0.08	0.03	0.05

Tabell 5.19: Denne tabellen viser testresultatet for $LBP_{8,1}$ og $VAR_{c8,1}$ benyttet på de originale cellekjerne-bildene i L23. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}. Den første forvirringsmatrisen er resultatet etter kombineringsen av TU-gruppe 6 og VAR_c -gruppe 4, mens den andre forvirringsmatrisen er resultatet etter kombineringsen av TU-gruppe 6 og VAR_c -gruppe 3.

	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	61 %
Gruppe 0:	29	19	48	Spesifisitet:	60 %
Gruppe 1:	7	12	19	Sensitivitet:	63 %
Totalt:	36	31	67	CCR _{eq} :	62 %
	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	67 %
Gruppe 0:	33	15	48	Spesifisitet:	69 %
Gruppe 1:	7	12	19	Sensitivitet:	63 %
Totalt:	40	27	67	CCR _{eq} :	66 %

5.18 kan vi imidlertid se at korrelasjonen er 0.58, og av den grunn velger vi å ikke utføre denne kombinasjonen.

5.6 Resultater fra bruk av operatorer benyttet på nabolag bestående av 16 piksler på originale bilder fra L23

Da vi benyttet disse operatorene på pasientene i M98, observerte vi at denne fremgangsmåten ikke førte til bedre resultater enn ved bruk av 8 piksels baserte operatorer på de nedskalerte cellekjerne-bildene. Vi skal nå se om dette endres ved å se på pasienter i L23. Vi har brukt fremgangsmåten i

Tabell 5.20: Tabellen viser resultatene etter trening og test i form av CCR_{eq}, etter bruk av LBP_{16,2} på pasienter i L23. I tillegg viser tabellen differansen mellom trening og test i prosentpoeng samt gjennomsnittlig trening- og testresultat.

TU-gr.:	1	2	3	4	5	6	7	8	...
Trening	49	55	64	49	51	62	53	59	...
Test	55	52	62	50	47	55	56	59	...
Differanse	-6	3	2	-1	4	7	-3	0	...

...	9	10	11	12	13	14	15	16	17	18	Gj.snitt
...	60	52	49	65	65	56	49	58	55	56	56 %
...	53	50	54	54	61	44	47	56	57	52	54 %
...	7	2	-5	11	4	12	2	2	-2	4	

seksjon 4.1.5 og kommet frem til følgende kvantisering:

$$\begin{aligned}
 0 &\rightarrow 1 \\
 1 &\rightarrow 2 \\
 2 &\rightarrow 3 \\
 3 &\rightarrow 4 \\
 4 &\rightarrow 5 \\
 5 &\rightarrow 6 \\
 6 &\rightarrow 7 \\
 7 - 9 &\rightarrow 8 \\
 8 - 11 &\rightarrow 9 \\
 > 11 &\rightarrow 10
 \end{aligned}$$

5.6.1 Resultater fra bruk av LBP

I tabell 5.20 kan vi se resultatene etter trening, i form av CCR_{eq}. Her er det TU-gruppe 3, 11 og 12 som gjør det best i trening med CCR_{eq} på 64-65 %, mens TU-gruppe 1, 4 og 15 gjør det dårligst med CCR_{eq} på 49 %. Gjennomsnittlig CCR_{eq} ble 56 %. Resultatene i mer detalj kan bli funnet i tabell 7.17 i Vedlegg C.

Resultatene etter test er å finne i tabell 5.20, i form av CCR_{eq}, mens forvirringsmatrisene og samtlige tilhørende beregninger kan bli funnet i tabell 7.18 i Vedlegg C. Etter test observerer vi at TU-gruppe 3 gjør det best med en CCR_{eq} på 62 %, mens TU-gruppe 14 gjør det dårligst med en CCR_{eq} på 44 %. Gjennomsnittlig CCR_{eq} ble 54 %, noe som innebærer et fall på kun 2 prosentpoeng fra trening.

5.6.2 Resultater fra bruk av varians

Resultater etter trening er å finne i tabell 5.21, hvor vi observerer at VAR_c-gruppe 2 oppnår best resultat med CCR_{eq} på 59 %, mens VAR_c-gruppe 1 og 3 gjør det dårligst med CCR_{eq} på 52 %. Gjennomsnittlig CCR_{eq} ble 55 %. Mer detaljerte resultater er å finne i tabell 7.19 i Vedlegg C.

Tabell 5.21: Tabellen viser resultatene etter trening og test i form av CCReq, etter bruk av $VAR_{c16,2}$ på pasienter i L23. I tillegg viser tabellen differansen mellom trening og test i prosentpoeng samt gjennomsnittlig trening- og testresultat.

TU-gr.:	1	2	3	4	5	6	7	8	9	10	Gj.snitt
Trening	52	59	52	54	56	55	57	53	54	58	55 %
Test	50	46	52	53	52	47	50	18	0	11	38 %
Differanse	2	13	0	1	4	8	7	35	54	47	

Testresultatene kan bli funnet i tabell 5.21, samt tabell 7.20 i Vedlegg C. Best ut kommer VAR_c -gruppe 4 med CCReq på 53 %, mens VAR_c -gruppe 9 kommer dårligst ut med CCReq på 0 %. Denne VAR_c -gruppen forekommer veldig sjeldent, og det er tilfeldigheter som gjør at det er forekomster av denne gruppen i trening, men ikke på test. Gjennomsnittlig CCReq ble 38 %, noe som er et fall på 17 prosentpoeng fra test.

5.6.3 Resultater etter kombinerings

Ved kombinasjon av $LBP_{16,2}$ og $VAR_{c16,2}$, observerte vi i tabell 5.20 at TU-gruppe 13 var en av de beste TU-gruppene, mens i tabell 5.21 observerte vi at VAR_c -gruppe 4 var en av de beste VAR_c -gruppene. Dersom vi ser på korrelasjonsmatrisen i tabell 5.22, observerer vi at en kombinasjon av TU-gruppe 13 og VAR_c -gruppe 4 har en korrelasjon på 0.1, og vi anser ikke egenskapene som for korrelerte. Vi lar med andre ord kombinasjonen bestå av TU-gruppe 13 og VAR_c -gruppe 4. I tabell 5.24 finner man resultatene etter test, og som man kan se oppnår vi CCReq på 60 %.

I likhet med tidligere ser vi også nærmere på kombinasjon av to TU-grupper som baseres på henholdsvis $LBP_{16,2}$ og $LBP_{8,1}$ i originalbildene. Resultatene fra $LBP_{8,1}$ i originalbildene viste at TU-gruppe 6 gjorde det bra i både trening og test, og av den grunn anser vi denne TU-gruppen som en god kandidat fra $LBP_{8,1}$. Tidligere i denne seksjonen observerte vi at tre TU-grupper endte opp med CCReq på over 60 % i test, men at kun to av disse gjorde det omtrent likt i trening. Dersom man regner ut korrelasjonskoeffisienten, $korr$, for kombinasjonen TU-gruppe 6 fra $LBP_{8,1}$ og TU-gruppe 3 fra $LBP_{16,2}$ får man $korr = 0.52$, mens man oppnår $korr = 0.04$ for kombinasjonen TU-gruppe 6 fra $LBP_{8,1}$ og TU-gruppe 13 fra $LBP_{16,2}$. Med andre ord vil kombinasjonen mellom TU-gruppe 6 fra $LBP_{8,1}$ og TU-gruppe 13 fra $LBP_{16,2}$ ha mest for seg. Resultatet etter kombinasjonen er å finne i tabell 5.23, og vi ender opp med en CCReq på 59 %.

Tabell 5.22: Korrelasjonsmatrise beregnet fra treningsettet til de originale bildene i L23. Her er det benyttet operatører med 16 piksels nabolag.

$VAR_c \rightarrow$ $TU \downarrow$	1	2	3	4	5	6	7	8	9	10
1	0.14	0.11	0.1	0.06	0.07	0.08	0.08	0.02	0.03	0.09
2	0.06	0	0.04	0.02	0.03	0.03	0.02	0.01	0.03	0
3	0.28	0.18	0.32	0.36	0.31	0.25	0.17	0.13	0.05	0.03
4	0.15	0.15	0.09	0.02	0.02	0.01	0.02	0.03	0.05	0.11
5	0.01	0.09	0.04	0.02	0.03	0.04	0.05	0.02	0.01	0.01
6	0.24	0.36	0.27	0.19	0.11	0.05	0.01	0	0.03	0.05
7	0.31	0.33	0.35	0.35	0.3	0.23	0.14	0.07	0.07	0.11
8	0.1	0.05	0.16	0.23	0.2	0.16	0.09	0.07	0.04	0.08
9	0.05	0.06	0.01	0.04	0.02	0.02	0.01	0.01	0.01	0.04
10	0.13	0.08	0.11	0.08	0.1	0.09	0.1	0.04	0.06	0.17
11	0.09	0.03	0.07	0.05	0.06	0.05	0.05	0.03	0.13	0.2
12	0.08	0.17	0.1	0.09	0.06	0.05	0.04	0.08	0.15	0.21
13	0.45	0.44	0.46	0.4	0.35	0.31	0.28	0.23	0.21	0.12
14	0.39	0.37	0.42	0.37	0.3	0.26	0.23	0.2	0.1	0.08
15	0.08	0.15	0.1	0.08	0.03	0.01	0.01	0.04	0.01	0.05
16	0.12	0.03	0.11	0.09	0.11	0.11	0.1	0.03	0.04	0.1
17	0.67	0.66	0.68	0.56	0.45	0.38	0.28	0.18	0.08	0.01
18	0.74	0.59	0.71	0.62	0.52	0.44	0.35	0.27	0.16	0.02

Tabell 5.23: Denne tabellen viser testresultatet etter kombinerings mellom TU-gruppe 6 fra LBP_{8,1} og TU-gruppe 13 fra LBP_{16,2}. Begge operatørene er benyttet på originalbildene.

	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	66 %
Gruppe 0:	36	12	48	Spesifisitet:	75 %
Gruppe 1:	11	8	19	Sensitivitet:	42 %
Totalt:	47	20	67	CCReq:	59 %

Tabell 5.24: Denne tabellen viser testresultatet etter kombinerings mellom TU-gruppe 13 og VARc-gruppe 4. Det er her brukt operatører med 16 piksels nabolag på de originale bildene.

	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	61 %
Gruppe 0:	30	18	48	Spesifisitet:	63 %
Gruppe 1:	8	11	19	Sensitivitet:	58 %
Totalt:	38	29	67	CCReq:	60 %

5.7 Resultater fra bruk av operatører benyttet på nabolag bestående av 8 piksler på nedskalerte bilder fra L23

Da vi så på resultatene for bruk av de ulike operatorene på M98, var det benyttelse av $LBP_{8,1}$ og $VAR_{c8,1}$ på nedskalerte bilder som oppnådde best resultater. På den annen side observerte vi, i likhet med de andre fremgangsmåtene, her en tendens til at resultatene snudde under test. Ved bruk av $LBP_{8,1}$ og $VAR_{c8,1}$ på de originale bildene fra L23, observerte vi derimot at resultatene ikke falt sammen på test. Av disse grunnene er det ikke utenkelig at $LBP_{8,1}$ og $VAR_{c8,1}$ benyttet på nedskalerte bilder fra L23 vil oppnå svært gode resultater. Vi har brukt fremgangsmåten i seksjon 4.1.5 for å kvantisere varians, og oppnådde følgende kvantisering:

$$\begin{aligned} 0 &\rightarrow 1 \\ 1 &\rightarrow 2 \\ 2 &\rightarrow 3 \\ 3 &\rightarrow 4 \\ 4 - 6 &\rightarrow 5 \\ 7 - 10 &\rightarrow 6 \\ 11 - 15 &\rightarrow 7 \\ 16 - 25 &\rightarrow 8 \\ 26 - 46 &\rightarrow 9 \\ > 46 &\rightarrow 10 \end{aligned}$$

5.7.1 Resultater fra bruk av LBP

Vi ser først på resultatene av $LBP_{8,1}$ brukt på treningsettet. I tabell 5.25 finner man alle resultatene fra trening oppgitt som CCReq, og mer utfyllende resultater er å finne i tabell 7.21 i Vedlegg C. Her observerer vi at TU-gruppe 7 gjør det best på trening med CCReq på 77 %. Den gruppen som gjør det dårligst er TU-gruppe 4 med CCReq på 51 %. Vi er her, som i tilfellet med nedskalerte bilder i M98, nødt til å se bort fra TU-gruppe 17 som har en CCReq på 0 %. Grunnen til dette er den samme som for M98, nemlig at det er for få forekomster av TU-gruppe 17 til at det er mulig å estimere sannsynlighetsfordelinger. Gjennomsnittlig CCReq ble 60 %.

Hvis vi nå tar for oss testresultatene, observerer vi at flere TU-grupper gjorde det bra. Det beste resultatet ble oppnådd av TU-gruppe 2 med en CCReq på 75 %, men som man kan se i tabell 5.25 er det flere TU-grupper som gjør det omtrent like bra. Disse gruppene er TU-gruppe 6-8 med en CCReq på 74 %. Det er verdt å merke seg at alle disse 4 gruppene gjør det bra på både trening og test. Dette er veldig lovende resultater. Den TU-gruppen som gjorde det dårligst, sett bort fra TU-gruppe 17, var TU-gruppe 15 med CCReq på 36 %. Gjennomsnittlig CCReq ble 54 %, hvilket gir et

Tabell 5.25: Denne tabellen viser trening- og testresultatet for $LBP_{8,1}$ benyttet på de nedskalerte cellekjerne-bildene i L23. Resultatene er oppgitt som CCReq i prosent. I tillegg viser tabellen differansen mellom trening og test i prosentpoeng samt gjennomsnittlig trening- og testresultat.

TU-gruppe	1	2	3	4	5	6	7	...
Trening	57	71	65	51	67	72	77	...
Test	53	75	52	46	56	74	74	...
Differanse	4	-4	13	5	11	-2	3	...

...	8	9	10	11	12	13	14	15	16	17	Gj.snitt
...	73	61	56	52	56	68	64	68	55	0	60 %
...	74	67	49	48	49	59	55	36	54	0	54 %
...	-1	-6	7	4	7	9	9	32	1	0	

Tabell 5.26: Denne tabellen viser trening- og testresultatet for VAR_c benyttet på de nedskalerte cellekjerne-bildene i L23. Resultatene er oppgitt som CCReq i prosent. I tillegg viser tabellen differansen mellom trening og test i prosentpoeng samt gjennomsnittlig trening- og testresultat.

VAR_c -gruppe	1	2	3	4	5	...
Trening	73	53	63	60	72	...
Test	79	44	54	56	58	...
Differanse	-6	9	9	4	14	...

...	6	7	8	9	10	Gj.snitt
...	73	74	65	57	58	65 %
...	64	62	65	59	55	60 %
...	9	12	0	-2	3	

fall på 6 prosentpoeng fra trening til test. Mer utfyllende resultater omkring test er å finne i tabell 7.22 i Vedlegg C.

5.7.2 Resultater fra bruk av varians

Dersom vi nå tar for oss resultatene fra bruk av $VAR_{c8,1}$ på trening, kan vi fra tabell 5.26 observere at VAR_c -gruppe 7 gjør det best med en CCReq på 74 %. Det dårligste resultatet ble oppnådd med VAR_c -gruppe 2 hvor CCReq kom på 53 %. Gjennomsnittlig CCReq ble 65 %. Mer utfyllende resultater er å finne i tabell 7.23 i Vedlegg C.

Vi går nå over til testresultatene etter bruk av $VAR_{c8,1}$. I tabell 5.26 observerer vi at VAR_c -gruppe 1 gjør det best med CCReq på 79 %. Dette er også et lovende resultat, selv om det kan være litt tilfeldig ettersom gruppe 1 kun oppnådde en CCReq på 73 % i trening. VAR_c -gruppe 2 oppnår dårligst resultat på test, med en CCReq på 44 %. Gjennomsnittlig CCReq ble 60 %, noe som er et fall på 5 prosentpoeng fra trening. I tabell 7.24 i Vedlegg C kan man finne mer utfyllende resultater.

Tabell 5.27: Korrelasjonsmatrise beregnet fra treningssettet til de nedskalerte bildene i L23. Her er det benyttet operatører med 8 piksels nabolag.

$VAR_c \rightarrow$ $TU \downarrow$	1	2	3	4	5	6	7	8	9	10
1	0.39	0.08	0.08	0.43	0.65	0.64	0.53	0.38	0.24	0.24
2	0.99	0.03	0.06	0.21	0.5	0.74	0.86	0.74	0.48	0.35
3	0.44	0.07	0.12	0.48	0.71	0.7	0.59	0.4	0.24	0.25
4	0.51	0.14	0.01	0.32	0.64	0.71	0.66	0.45	0.25	0.27
5	0.86	0.1	0.05	0.3	0.66	0.89	0.93	0.67	0.34	0.22
6	0.9	0.13	0.02	0.22	0.58	0.82	0.93	0.78	0.47	0.3
7	0.87	0.13	0.06	0.14	0.46	0.7	0.84	0.82	0.61	0.38
8	0.75	0.07	0.01	0.17	0.37	0.5	0.68	0.85	0.79	0.62
9	0.6	0.02	0.03	0.11	0.32	0.44	0.58	0.74	0.67	0.58
10	0.35	0.06	0.1	0.45	0.67	0.61	0.48	0.34	0.24	0.25
11	0.41	0.07	0.13	0.5	0.73	0.71	0.56	0.36	0.22	0.23
12	0.41	0.12	0.07	0.44	0.68	0.68	0.55	0.35	0.19	0.18
13	0.54	0.1	0.06	0.37	0.64	0.71	0.67	0.49	0.29	0.23
14	0.05	0.05	0.07	0.11	0.1	0.08	0.04	0.06	0.1	0.33
15	0.26	0.05	0.01	0.22	0.37	0.39	0.39	0.38	0.31	0.26
16	0.3	0.05	0.08	0.38	0.55	0.5	0.42	0.31	0.21	0.22
17	-	-	-	-	-	-	-	-	-	-

5.7.3 Resultater etter kombinerings

Ved kombinerings kan vi først observere at TU-gruppe 7 gjør det best på trening, og oppnår også svært gode resultater på test. Av den grunn er denne gruppen et naturlig valg til kombinerings. Dersom vi ser på VAR_c -gruppene, observerer vi at VAR_c -gruppe 7 gjør det best på trening, men etter test gjør denne gruppen det 12 prosentpoeng dårligere. Vi kan derimot se at VAR_c -gruppe 1 gjør det bra i både trening og test, men korrelasjonen mellom TU-gruppe 7 og VAR_c -gruppe 1 er 0.87, se tabell 5.27. Med andre ord vil ikke denne kombinasjonen være et godt valg. Selv om VAR_c -gruppe 3 ikke er blant de beste i hverken trening eller test, observerer vi at denne gruppen har veldig lav korrelasjon til TU-gruppe 7. Vi velger derfor å kombinere TU-gruppe 7 med VAR_c -gruppe 3. Resultatene etter kombinerings er å finne i tabell 5.28, og CReq kom på 72 %.

Da vi jobbet med de nedskalerte bildene i M98, så vi også på muligheten til å kombinere en TU-gruppe funnet i de originale cellekjerne-bildene, med samme TU-gruppe funnet i de nedskalerte cellekjerne-bildene. Vi velger å gjøre tilsvarende her. Hvis vi ser på resultatene til de ulike TU-gruppene, observerer vi at TU-gruppe 17 som ble funnet i de originale cellekjerne-bildene gjør det bra i trening og test, se tabell 5.16. Dersom vi ser på tabell 5.25, observerer vi at TU-gruppe 7, beregnet fra de nedskalerte cellekjerne-bildene, gjør det bra i både trening og test. Korrelasjonskoeffisienten mellom disse to gruppene ble 0.08, så vi velger å benytte oss av denne kombinasjonen. Resultatet etter kombinerings er å finne i tabell 5.29, og som vi kan se ble resultatet 65 %.

Tabell 5.28: Resultatet etter kombineringsen av TU-gruppe 7 og VAR_c -gruppe 3. Egenskapene er funnet ved å se på de nedskalerte cellekjerne-bildene i L23.

	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	31	17	48	Spesifisitet:	65 %
Gruppe 1:	4	15	19	Sensitivitet:	79 %
Totalt:	35	32	67	CCReq:	72 %

Tabell 5.29: Resultatet etter kombineringsen av TU-gruppe 7 funnet fra de nedskalerte cellekjerne-bildene i L23 og TU-gruppe 17 funnet fra de originale bildene i L23. Det har blitt benyttet operatører med 8 piksels nabolag i begge tilfeller.

	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	54 %
Gruppe 0:	19	29	48	Spesifisitet:	40 %
Gruppe 1:	2	17	19	Sensitivitet:	89 %
Totalt:	21	46	67	CCReq:	65 %

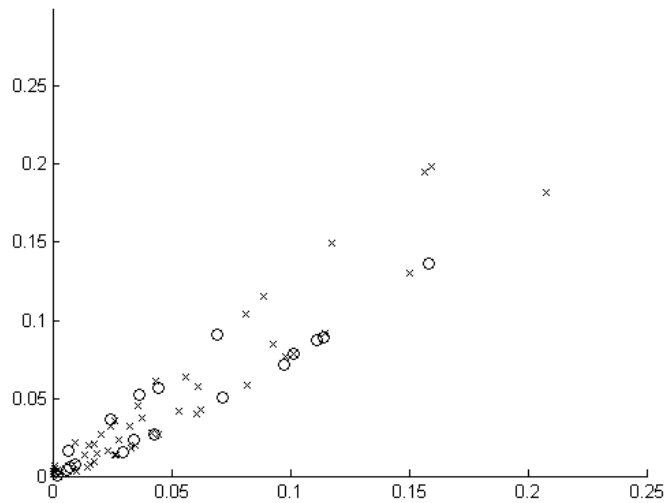
5.8 Diskusjon rundt datasettet L23

Resultatene etter bruk av LBP og VAR_c på L23 ble i flere tilfeller forholdsvis gode, og i motsetning til i M98 forekom dette i både trening og test. I denne seksjonen starter vi med å se på $LBP_{8,1}$ og $VAR_{c8,1}$ benyttet på originalbildene, før vi diskuterer bruken av $LBP_{16,2}$ og $VAR_{c16,2}$ på de originale cellekjerne-bildene. Deretter går videre med å se på $LBP_{8,1}$ og $VAR_{c8,1}$ på de nedskalerte bildene. Seksjonen avsluttes med en mer generell diskusjon omkring det vi har oppnådd med datasettet L23.

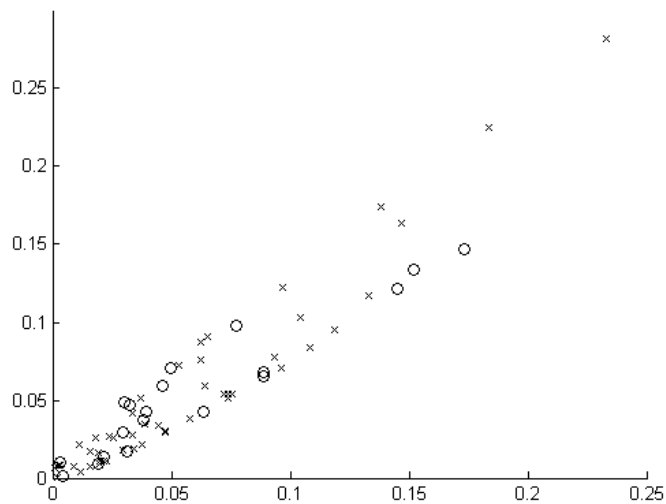
5.8.1 Operatører benyttet på 8 piksles nabolag i originalbildene

I seksjon 5.5.1 observerte vi at bruken av $LBP_{8,1}$ på originalbildene i L23 førte til mange resultater rundt 50 %, og noen over 60 % ved testing. Dette er mye bedre resultater enn det vi oppnådde med tilsvarende fremgangsmåte på M98. Vi kan nå se at det er likheter mellom trening og test, noe vi ikke var i nærheten av å se i M98. Ved å se på spredningsplottene til TU-gruppe 9, kan vi observere at det er samsvar mellom trening og test i hvordan pasientene fra de ulike prognosegruppene forekommer. Disse plottene er å finne i figur 5.9. Som vi kan se er det pasientene med god prognose som har mest varierende distanser. I flere tilfeller observerer vi at pasientene med dårlig prognose forekommer oftest med relativt lave distanser sammenliknet med pasientene med god prognose. Dette er tilfellet for blant annet TU-gruppe 14, og er illustrert i figur 5.10a. Pasientene med god prognose har både høye og lave distanser, og dominerer med andre ord de høye distansene. På denne måten kan man i flere tilfeller oppnå en forholdsvis god klassifisering ved å markere pasienter med lave distanser som pasienter med dårlig prognose, og motsatt. Det er imidlertid verdt å merke seg at en skillelinje som står 135 ° på x-aksen ikke nødvendigvis er den optimale i alle tilfeller. I figur 5.10b kan vi for eksempel se en annen linje som er langt mer optimal for å oppnå høyest mulig CCReq.

Dersom vi ser på testresultatene, som i beste tilfelle ble 63 %, kan vi ikke si at dette er veldig tilfredsstillende resultater. Vi må imidlertid huske på



(a) Trening



(b) Test

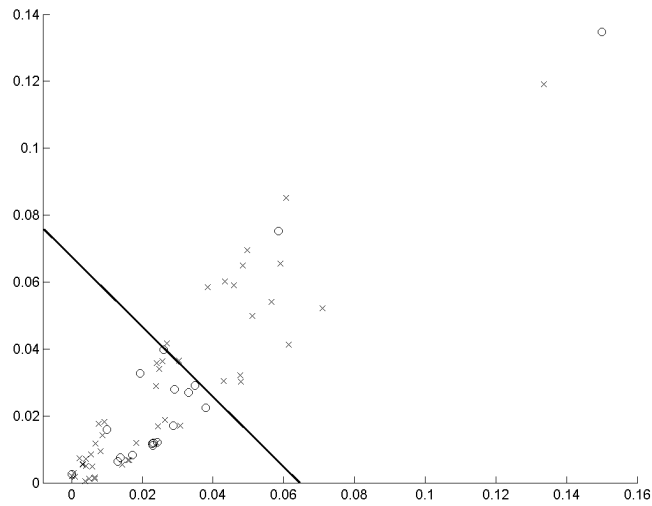
Figur 5.9: Spredningsplott basert på distansene fra TU-gruppe 9, for både trening og test. X representerer god prognose, mens O representerer dårlig prognose. Distanse til klassen med god prognose er vist på x -aksen, mens y -aksen viser pasientens distanse til klassen med dårlig prognose.

at tidligere klassifiseringer benyttet på dette materialet har gitt resultater rundt 70 %. Nielsen og Danielsen oppnådde blant annet en CCR på 70 % etter validering i [50]. Her var trening- og testsettet de benyttet seg av nesten helt likt vårt. Den eneste ulikheten er 3 pasienter som har endret prognosegruppe, etter gjennomgåelse av patologer på datasettet i senere tid. CCR er omtrent det samme som CCReq. Forskjellen er at CCR er et vektet gjennomsnitt mellom sensitivitet og spesifisitet, hvor vektene er andelen av hver gruppe. I [50] er andelen forekomster med god og dårlig prognose henholdsvis 30 % og 70 %. Ettersom de beste resultatene vi har sett på dette settet ligger rundt 70 %, kan man se at L23 er et vanskelig datasett å oppnå god klassifisering på.

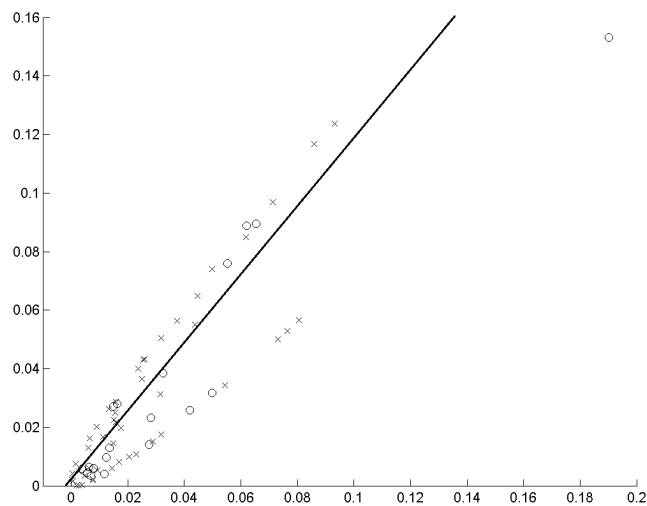
Hvis vi nå ser tilbake på resultatene vi oppnådde ved bruk av $VAR_{c8,1}$ på originalbildene, husker vi at det også her var samsvar mellom flere resultater i trening og test. I de aller fleste tilfellene lå CCReq fra test over 50 %. Det er imidlertid, som vi var inne på i seksjon 5.5.2, kun én VAR_c -gruppe som oppnår en CCReq på over 60 % i test. Det var VAR_c -gruppe 9 som oppnådde en CCReq på 61 %. Det er imidlertid et litt dårlig tegn at CCReq i trening for samme VAR_c -gruppe ble 56 %. Det tyder på at CCReq vi oppnår etter test ikke er helt reelt i det generelle tilfellet. Til tross for dette ser det ut til at det er klasseskillende informasjon å finne når man også ser på $VAR_{c8,1}$.

Vi kan observere at resultatene totalt sett viser mye mer samsvar mellom trening og test, i forhold til hva vi så for samme operatører i M98. I de fleste tilfeller ser det også ut til at det er klasseskillende informasjon å hente ved bruk av både $LBP_{8,1}$ og $VAR_{c8,1}$. Når vi sammenlikner resultatene fra L23 med M98, er det viktig å huske på at disse datasettene består av cellekjerne fra ulike typer kreft. I tillegg kan cellekjerne i de ulike settene være plukket ut av ulike personer, noe som gjør at datasettene også er påvirket av subjektive vurderinger. Dette vil igjen ha innvirkning på hvor gode resultatene etter klassifisering blir.

Da vi jobbet med M98 så vi at sannsynlighetsfordelingene ikke stemte overens i trening og test, når man forholdt seg til en prognosegruppe. Vi kan derimot se at dette er tilfellet i L23, og i figur 5.11 har vi en illustrasjon på det. Hvis vi starter med å se på de 4 sannsynlighetsfordelingene fra L23 i denne figuren, kan vi se at formen i sannsynlighetsfordelingene til pasientene med god prognose er veldig lik i både trening og test. Samtidig skiller disse seg fra formen i sannsynlighetsfordelingene til pasientene med dårlig prognose. Formen til sannsynlighetsfordelingene til pasientene med dårlig prognose bærer ikke den samme graden av likhet, men de har begge høyere varians enn fordelingene til pasientene med god prognose. Dette ser vi ikke var tilfellet i M98, og dette kan vi se fra tallene i tabell 5.30. Her observerer vi også at forventningsverdien til sannsynlighetsfordelingene fra god prognose i L23 er ulik de fra dårlig prognose. Vi kan nemlig se at forventningsverdiene til pasientene med god prognose er høyere enn tilsvarende forventningsverdi for dårlig prognose. I tillegg ser forventningsverdiene i trening og test ut til å være omtrent lik, når man ser på sannsynlighetsfordelingene til en prognose-



(a) TU-gruppe 12 - Trening



(b) TU-gruppe 4 - Test

Figur 5.10: Denne figuren viser tilnærming til optimal skillelinje mellom pasienter med god og dårlig prognose. X representerer god prognose, mens O representerer dårlig prognose. Spredningsplottet er basert på treningsettet i L23, for TU-gruppe 12 og testsettet i L23 for TU-gruppe 4.

gruppe. Det er derfor grunn til å tro at det er mulig å skille klassene ved å legge vekt på forventningsverdien til en sannsynlighetsfordeling når man skal måle distanser. Dette støtter med andre ord bruken av Mahalanobis distanse-mål. Selv om vi kun har sett på TU-gruppe 6, er det verdt å merke seg at forventningsverdiene oppfører seg likt i de to prognosegruppene, for alle TU-gruppene. Det er av den grunn ikke urimelig å tenke at vi i flere tilfeller kan oppnå minst like gode resultater ved bruk av Mahalanobis distanse som ved bruk av Gaussisk Bhattacharyya distanse.

Fra sannsynlighetsfordelingene til VAR_c ser vi at det er litt liknende tendenser som vi så for TU-gruppene. Hvis vi tar for oss sannsynlighetsfordelingene til VAR_c -gruppe 3, som er å finne i figur 5.12, kan vi se at det er størst likhet mellom sannsynlighetsfordelingene for gruppene med god prognose. Her er det imidlertid ikke like stor likhet mellom trening og test, som det vi så for TU-gruppe 6. I likhet med for TU, er det også sannsynlighetsfordelingen for dårlig prognose i trening som skiller seg mest ut fra de andre. Sannsynlighetsfordelingen til dårlig prognose i test er tilsynelatende mer lik sannsynlighetsfordelingene til pasientene med god prognose, fremfor sannsynlighetsfordelingen for dårlig prognose i trening. Ettersom sannsynlighetsfordelingene for god prognose er såpass like hverandre, ser det ut til at CCR_{eq} blir omtrent uendret fra trening til test. Dette kan sies om de fleste VAR_c -gruppene, men ikke alle. Det er verdt å merke seg at vi ikke ser et like tydelig skille mellom klassenes sannsynlighetsfordelinger for VAR_c -gruppene, som det vi gjorde for TU-gruppene. I tillegg er det verdt å påpeke at det er lite å hente av klasseskillende informasjon fra sannsynlighetsfordelingenes forventningsverdier. Ettersom flertallet av sannsynlighetsfordelingene er tilnærmet normalfordelte, støtter dette bruk av Gaussisk Bhattacharyya som distanse-måler mellom fordelingene.

Ved valg av klassifikator erfarte vi i M98 at det kun var noen få spredningsplott hvor pasientene dannet en parabel, mens resten kunne tilpasses normalfordeling. I L23 erfarte vi derimot at pasientene i over halvparten av spredningsplottene dannet en parabel. Av den grunn kunne det vært fordelaktig å velge en annen klassifikator enn multivariabel gaussisk klassifikator. Vi valgte derimot å bruke samme klassifikator som i M98 for å gjøre sammenlikning av resultatene enklere, og for å holde forsøkene konsistente.

Da vi kombinerte TU-gruppe 6, som gjorde det relativt bra i både trening og testing, med VAR_c -gruppe 4, som gjorde det bra i trening men falt litt igjen i test, endte vi opp med CCR_{eq} på 62 %. Dette er bare 1 prosentpoeng bedre enn det TU-gruppe 6 klarte å oppnå alene. Ettersom VAR_c -gruppe 4 falt igjennom på test, kan nettopp det være grunnen til at resultatene ikke ble så bra som vi hadde håpet. Ved kombinerings av den samme TU-gruppen med VAR_c -gruppe 3, endte vi opp med en CCR_{eq} på 66 %. VAR_c -gruppe 3 hadde et litt dårligere resultat etter trening, men viste seg å ikke falle igjennom på test. Vi fikk med andre ord en økning på 5 prosentpoeng fra å bare bruke TU-gruppe 6. Økningen er ikke overveldende høy, men vi kan se at kombineringsen har noe for seg. I motsetning til i M98, ser det ikke lenger

Tabell 5.30: Denne tabellen viser forventningsverdi og varians for TU-gruppe 6 på trening og test. Verdiene er utregnet både for M98 og L23.

Datasett	Prognose	μ -trening	μ -test	σ^2 -trening	σ^2 -test
M98	God	73	75	54.4	59.7
M98	Dårlig	73.5	72.2	54.7	54.1
L23	God	77.2	77.3	53.9	54.1
L23	Dårlig	76.2	76.1	58	55.9

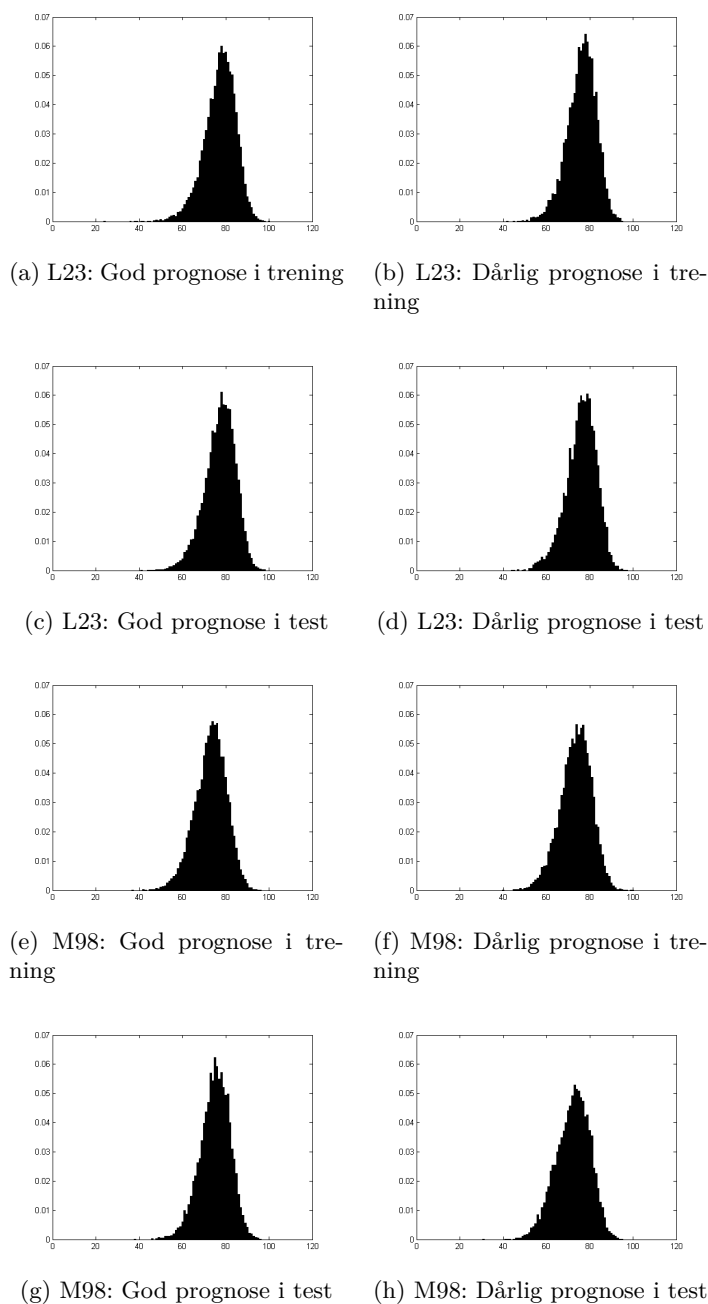
ut til at vi er et offer for ”curse of dimensionality”, takket være bruk av flere pasienter i de to datasettene. I tillegg er vi nå kun 4 prosentpoeng unna klassifiseringsraten Nielsen og Danielsen oppnådde på det samme datasettet i [50].

5.8.2 Operatører benyttet på 16 piksels nabolag i originale cellekjerne-bilder

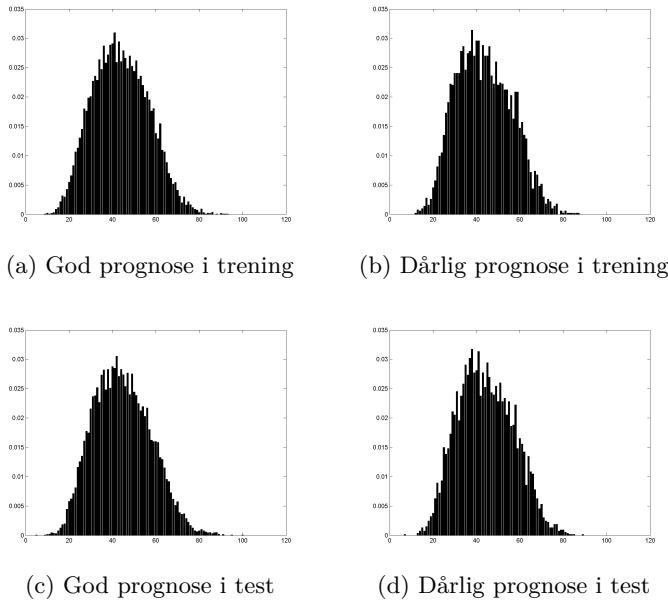
I seksjon 5.6, som omhandler resultatene fra bruk av $LBP_{16,2}$ og $VAR_{c16,2}$, så vi at resultatene ikke ble særlig bra. De fleste resultatene etter både trening og test lå litt over 50 %, og det var kun 3 TU-grupper som hadde CCReq på over 60 %. De 3 TU-gruppene som hadde resultater på over 60 % i test, hadde også forholdsvis gode resultater på treningsettet, hvilket er en indikator på at disse gruppene generelt inneholder klasseskillende informasjon. Disse resultatene er på omtrent lik linje som de vi oppnådde med $LBP_{8,1}$ på originalbildene, og støtter ikke opp om at bruken av operatører på en større andel av cellekjernen gir bedre resultater.

Selv om TU-gruppene ikke ga de beste resultatene, husker vi fra seksjon 5.6 at VAR_c -gruppene oppnådde enda dårligere resultater. Her observerte vi også at et resultat endte med CCReq på 0 %. Som vi skrev kommer dette av at det er svært få forekomster av denne VAR_c -gruppen, noe som gjør at den tilhørende sannsynlighetsfordelingen utgjør et svært dårlig estimat. I figur 5.13a kan vi se at andelen forekomster er veldig skjevt fordelt til tross for at kvantiseringen prøver å gjøre disse forekomstene like. I figur 5.13b finner vi fordelingene av VAR_c -verdiene før kvantisering, og her kan vi tydelig se at VAR_c -verdi lik 0 (1 i histogrammet) forekommer langt oftere enn alle de andre verdiene. Dette kan vi også observere hvis vi ser på en cellekjernes VAR_c -verdier, slik som er illustrert i figur 5.14. Med andre ord er det veldig lite varians i disse cellekjernene når man ser på 16 piksler, noe som gjør at vi har lite å hente klasseskillende informasjon ut fra.

I seksjon 5.6 så vi at kombinasjon mellom TU-gruppe 13 og VAR_c -gruppe 4 funnet med henholdsvis $LBP_{16,2}$ og $VAR_{c16,2}$ resulterte i CCReq på 60 %. Dette er bedre resultater enn det VAR_c -gruppe 4 klarte å oppnå alene, men dårligere enn det TU-gruppe 13 oppnådde, som var 61 %. Denne kombinasjonen hjelper med andre ord ikke på resultatet. Vi kombinerte også $LBP_{16,2}$ med $LBP_{8,1}$ og her endte vi opp med litt bedre resultater enn det én TU-



Figur 5.11: Sannsynlighetsfordelingene til TU-gruppe 6. a, b, c og d er god og dårlig prognose i henholdsvis trening og testsettet til L23. e, f, g og h er tilsvarende, men for M98



Figur 5.12: Sannsynlighetsfordelinger til VAR_c -gruppe 3. a, b, c og d for god og dårlig prognose i henholdsvis trening og testsett i L23.

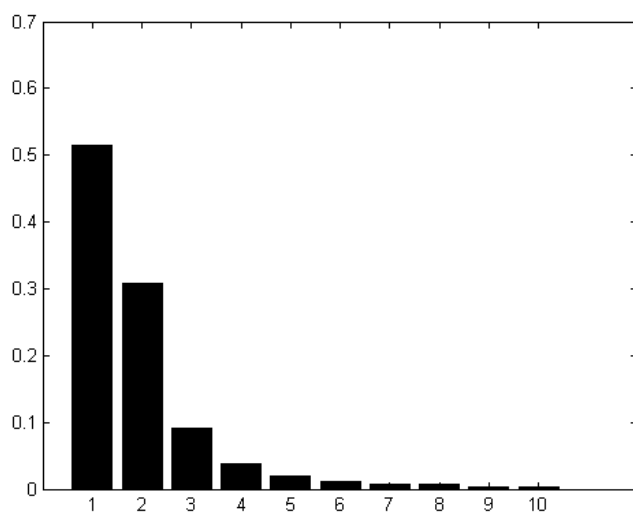
gruppe fra $LBP_{8,1}$ og $LBP_{16,2}$ klarte alene. Med det sagt, må vi imidlertid påpeke at kombinasjonen ikke oppnådde mye bedre resultater, og det kan være kun tilfeldigheter som gjorde at kombinasjonen gjorde det litt bedre.

5.8.3 Operatører benyttet på 8 piksles nabolag i nedskalerte cellekjerne-bilder

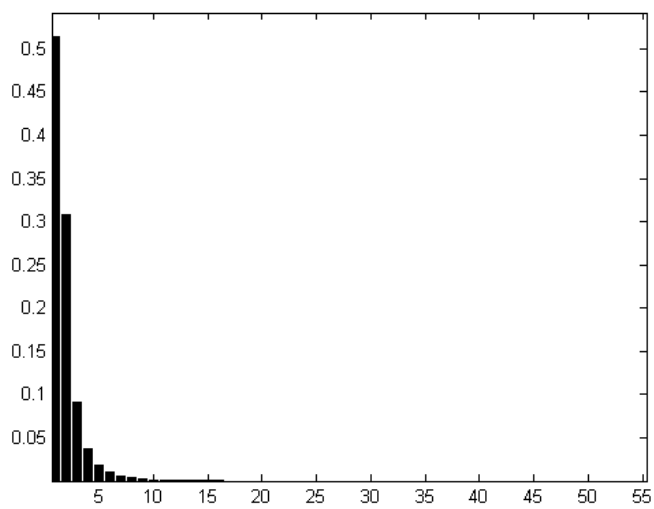
Det første vi legger merke til når vi ser på resultatene etter klassifisering basert på de nedskalerte cellekjerne-bildene, er at de er svært gode. Ved bruk av både $LBP_{8,1}$ og $VAR_{c8,1}$ ser vi at enkelte grupper oppnår CCReq på over 70 % i test. Det beste resultatet etter bruk av $LBP_{8,1}$ ble oppnådd av TU-gruppe 2, som fikk en CCReq på 75 %. Det beste resultatet etter bruk av $VAR_{c8,1}$ på test ble 79 %, men dette var en økning på 6 prosentpoeng fra trening (tabell 5.26). Det tyder på at det er et litt tilfeldig resultat, og at vi ikke helt kan stole på at resultatet er så godt som 79 % i et generelt tilfelle.

Resultatene er svært gode med tanke på at 70 % er det Nielsen og Danielsen oppnådde med adaptive egenskaper i [50]. Når vi ser på hvilke TU-grupper som oppnådde de beste resultatene, ser vi at disse stammer fra de homogene binære mønstrene. Blant disse har vi TU-gruppe 2, som kommer av helt flate områder i bildet, samt TU-gruppe 6, 7 og 8, som har løpelengde på minst 5 (med løpelengde tenker på vi hvor mange 0/1 som havner etter hverandre i det binære mønsteret før det motsatte tallet (1/0) forekommer).

Når vi ser på egenskapene som gjør det dårligst i test, ser vi at det også her er TU-grupper som dominerer. Verst ut kommer TU-gruppe 15, som

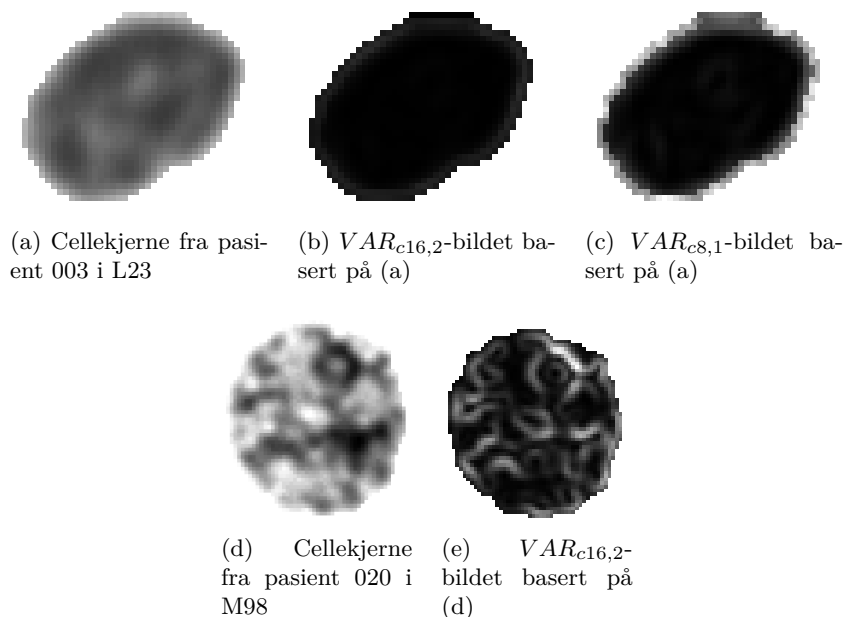


(a) Fordelingen av varians-forekomster etter kvantisering.



(b) Fordelingen av varians før kvantisering. Verdiene på x-aksen er lagt til 1 for å tilpasse array-indekseringen i MATLAB.

Figur 5.13: Fordelingn av varians før og etter kvantisering.



Figur 5.14: Cellekjerne-bilder fra pasienter med god prognose fra henholdsvis L23 og M98, samt tilhørende VAR_c -bilder.

faller fra 68 % i trening til 36 % i test, og viser en tydelig tendens til overtilpassning. Dette tyder på at det kan eksistere klasseskillende informasjon her, men at denne ikke kommer frem på grunn av dårlig samsvar mellom forekomstene i trening og test. Her er det verdt å merke seg at selv om vi har flere pasienter enn da vi så på M98, stammer TU-gruppe 15 fra de ikke-uniforme mønstrene. Som Ojala et al. [54] har påpekt forekommer disse svært sjeldent, sammenliknet med de uniforme mønstrene. Dette var også tilfellet i figur 4.4, som viste andelen forekomster av de 36 ulike mønstrene.

Vi kan derimot si med større sikkerhet at TU-gruppe 1, 4, 10, 11, 12 og 16 ikke inneholder så mye klasseskillende informasjon. Disse gruppene ender opp med resultater sentrert rundt 50 % i både trening og test, og som vi kan se er fire av disse ikke-uniforme mønstre. De to uniforme mønstrene, altså TU-gruppe 1 og 4, er de to mønstrene som har lengst løpelengde bestående av enere. Med andre ord består de to mønstrene av henholdsvis en piksel omgitt av mørkere piksler rundt, og en piksel med kun en nabo med mørkere gråtone.

Når vi ser på VAR_c -gruppene, observerer vi fra tabell 5.26 at det kun er en VAR_c -gruppe som resulterer i mindre enn 50 % på test. VAR_c -gruppe 2 ser ut til å være blant gruppene hvor det er minst å hente av klasseskillende informasjon. Vi kan også se at VAR_c -gruppe 5 er gruppen som faller mest igjennom fra trening til test, med en fallhøyde på hele 14 prosentpoeng. Med dette sagt, ser vi at det er 4 grupper som holder seg mer eller mindre stabilt når man går fra trening til test.

Ved å sette resultatene fra operatorer med 8 piksels nabolag i originalbildene opp mot resultatene vi har sett på i denne seksjonen, ser vi at TU-gruppe 6 går igjen med gode resultater i både trening og test. Med andre ord ser det ut til at mye av ulikheten mellom god og dårlig prognose, ligger i hvor ofte en piksel har kun 2 naboer med mørkere gråtoner. Det er verdt å merke seg at målet etter andelen med 2 mørkere naboer, er av langt høyere prognostisk verdi, enn når man kun måler andel av forekomster hvor kun 1 nabo er mørkere. Dette kan vi se når vi sammenlikner resultatene fra TU-gruppe 6 og 4.

I seksjon 5.7.3 ser vi fra resultatene etter kombineringsresultatene ga lite. Ved å kombinere TU-gruppe 7 med VAR_c -gruppe 1, ser vi at CCReq kom på 75 %, noe som er tilsvarende det TU-gruppe 2 oppnådde alene. Vi har tidligere sett at kombineringsresultatene av TU og VAR_c har gitt bedre resultater, noe som tyder på at vi ikke nødvendigvis er offer for "curse of dimensionality". Det kan i stedet tenkes at TU-gruppen og VAR_c -gruppen vi ser på overlapper hverandre for mye ved separering av prognosegruppene, selv om korrelasjonen mellom egenskapene var veldig lav. Det tyder dermed på at kombineringsresultatene mellom TU og VAR_c har lite for seg i dette tilfellet.

Ved kombineringsresultatene av distanser fra TU-gruppe 6, men fra ulike oppløsninger på bildene, så vi at heller ikke dette resulterte i en god CCReq. Her kom CCReq på 73%, noe som er dårligere enn det TU-gruppe 6 klarer alene på de nedskalerte cellekjerne-bildene.

5.8.4 Oppsummering av hva vi har sett ved analyse av L23

For å ta en kort oppsummering, har vi i likhet med M98 sett at de beste resultatene oppnås ved bruk av 8 piksels operatorer på nedskalerte cellekjerne-bilder. Vi har opplevd at det å bruke flere pasienter medfører mer samsvar mellom resultatene i trening og test, enn det vi så i M98. Dette tyder på at fremgangsmåten krever en del treningsdata, men til gjengjeld ser vi at LBP klarer å finne klaseskillende informasjon mellom god og dårlig prognose.

Det er verdt å merke seg at det faktisk er forskjeller mellom cellekjerne-ne i L23 og cellekjernene i M98. Dette merket vi da vi jobbet med $VAR_{c16,2}$ på originalbildene. Her så vi at cellekjernene i L23 inneholder langt mindre variasjon enn cellekjernene i M98. Av den grunn endte vi ikke opp med veldig gode resultater ved bruk av denne operatoren på cellekjerne-bildene i L23, men som vi var inne på kan man forvente bedre resultater om man benytter seg av en annen kvantisering av VAR_c -verdiene.

Vi har også sett at TU-gruppe 6 oppnår gode resultater ved bruk av $LBP_{8,1}$ benyttet på både originale og nedskalerte cellekjerne-bilder i L23. For $LBP_{16,2}$ tilsvarer det TU-gruppe 6 eller 10. Dette avhenger av om man teller antall 0 eller om man ser på andelen 0 i det binære mønsteret. Vi observerte imidlertid at ingen av disse TU-gruppene var med på å gi de beste resultatene ved bruk av $LBP_{16,2}$, så suksessen fra mønsteret til TU-gruppe

6 gjelder kun ved bruk av $LBP_{8,1}$.

5.9 Resultater fra bruk av adaptive egenskaper benyttet på nabolag bestående av 8 piksler på originale bilder fra M98

I denne seksjonen presenteres trening- og testresultater etter bruk av de adaptive egenskapene beskrevet i 4.4.

5.9.1 Kombinasjon av TU og entropi

Ved trening resulterte bruk av F_+ i en CCR_{eq} på 70 %, noe som ser lovende ut. Ved klassifisering av testsettet oppnår vi en CCR_{eq} på 29 %. Dette er ikke gode resultater, og vi observerer at resultatene vi oppnådde med adaptive egenskaper opptrer på samme måte som ved bruk av $LBP_{8,1}$ på det samme datasettet. Dette gir oss nok en grunn til å tro at de dårlige testresultatene kommer av for få pasienter i datasettet.

5.9.2 Kombinasjon av TU og varians

Ved bruk av adaptive egenskaper som kombinerer TU og VAR_c , satt vi først opp en korrelasjonsmatrise for å se hvor like F_+ og F_- er, slik som ved bruk av entropi. Dette gjør vi for å kunne avgjøre om vi skal benytte både F_+ og F_- , eller kun én av egenskapene. Korrelasjonsmatrisen ble

	F_+ -god	F_+ -dårlig	F_- -god	F_- -dårlig
F_+ -god	1.0	0.9	1.0	0.9
F_+ -dårlig	0.9	1.0	0.9	1.0
F_- -god	1.0	0.9	1.0	0.9
F_- -dårlig	0.9	1.0	0.9	1.0

Som vi kan se fra matrisen er F_+ og F_- helt overlappende, og av den grunn velger vi å kun benytte F_+ . Litt mer urovekkende er at korrelasjonen mellom god og dårlig prognose er svært høy. På trening oppnådde vi en CCR_{eq} på 60 %, men på test gikk CCR_{eq} ned til 25 %. Dette er ikke gode resultater, men det er i samsvar med hva vi har sett tidligere på M98.

5.9.3 Kombinasjon av TU og senterpikselens gråtone

Ettersom de to andre kombinasjonene resulterte i såpass lave CCR_{eq} etter test, og dette har vært gjennomgående i bruken av M98 sine supertreningsett, kommer vi ikke til å utføre kjøring med adaptive egenskaper basert på TU-grupper og senterpikselens gråtone på disse supertreningsettene.

Tabell 5.31: Tabellen viser resultatene etter trening i form av CCR_{eq} i prosent, etter bruk av adaptive egenskaper på pasienter fra M98. De adaptive egenskapene er en kombinasjon av $LBP_{8,1}$ og $ENT_{8,1}$.

F_+	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	0	10	10	Sensitivitet:	100 %
Totalt:	4	16	20	CCR _{eq} :	70 %

Tabell 5.32: Tabellen viser resultatene etter testing i form av CCR_{eq} i prosent, etter bruk av adaptive egenskaper på pasienter fra M98. De adaptive egenskapene er en kombinasjon av LBP_{8,1} og ENT_{8,1}.

F_+	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	4	2	6	Sensitivitet:	33 %
Totalt:	6	8	14	CCR _{eq} :	29 %

Tabell 5.33: Resultater etter klassifisering av treningsettet i M98. Resultatene kommer fra bruk av adaptive metoder med varians.

F_+	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	5	5	10	Spesifisitet:	50 %
Gruppe 1:	3	7	10	Sensitivitet:	70 %
Totalt:	8	12	20	CCR _{eq} :	60 %

Tabell 5.34: Resultater etter klassifisering av testsettet i M98. Resultatene kommer fra bruk av adaptive metoder med varians.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	21 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	3	3	6	Sensitivitet:	50 %
Totalt:	3	11	14	CCR _{eq} :	25 %

5.10 Diskusjon rundt adaptive egenskaper benyttet på M98

I denne seksjonen diskuterer vi resultatene vi oppnådde ved bruk av adaptive egenskaper på M98. Først diskuterer vi kombinasjonen av TU og entropi, for deretter å diskutere kombinasjonen av TU og varians.

5.10.1 Kombinasjon mellom TU og entropi

Etter bruk av adaptive egenskaper på M98, erfarte vi det samme som ved bruk av $LBP_{8,1}$ og VAR_c . Vi oppnådde greie resultater på trening (70 %), for så å få omtrent helt motsatte resultater på test (29 %). Igjen kan dette kan komme av for få pasienter i M98. For å oppnå bedre resultater kunne det vært en mulighet å brukt et større vindu for å beregne entropi. I tabell 5.35, observerer vi at entropien er dominert av forekomster av entropi-gruppe 10. Ved å bruke et større vindu får man flere entropi-verdier, og forhåpentligvis litt bedre spredning av forekomstene. Dette kan igjen gi opphav til mer klasseskillende informasjon, og med det bedre klassifiseringsresultater.

Ved beregning av entropi bruker vi Shannon entropi, men det hadde vært mulig å bruke en annen utregning for entropi, og med det muligens oppnå bedre klassifiseringsresultater. Shannon entropi er kun et spesialtilfelle av Rényi entropi [1], som er gitt ved

$$H_{\alpha}(X) = \frac{1}{1-\alpha} \log\left(\sum_{i=1}^n p_i^{\alpha}\right)$$

Shannon entropi oppnår man når $\alpha \rightarrow 1$. Det kan tenkes at det å bruke en annen verdi for α ville ført til bedre resultater. Vi så i tabell 4.4 at vi endte opp med 21 ulike entropiverdier ved bruk av Shannon entropi, selv om det er 22 ulike tilfeller totalt når man ser på 8 piksler. Hadde vi benyttet et annet mål på entropi, kunne vi kanskje endt opp med 22 ulike tilfeller, og på den måten oppnådd en litt bedre spredning av verdiene. Nå har det seg slik at entropi-verdi lik 3 forekommer i over 70 % i alle cellekjerne-bilder i treningssettet i M98. Bruk av en annen α kan føre til at kvantiseringen av entropien ville medført en litt bedre spredning, men ettersom entropi-verdi lik 3 forekommer såpass ofte, er det ikke sikkert at dette ville gitt så mye bedre resultater.

Til nå har vi drøftet vår måling av entropi, men det er ikke nødvendigvis det eneste steget i fremgangsmåten som kunne blitt gjort anderledes for å oppnå bedre resultater. Etter at entropi er utregnet, og de adaptive egenskapene er funnet, valgte vi å sette opp sannsynlighetsfordelinger over hva de adaptive egenskapene ble i hvert cellekjerne-bilde. Disse sannsynlighetsfordelingene satte vi opp for hver pasient, samt hver prognosegruppe. Deretter beregnet vi den Gaussiske Bhattacharyya avstanden fra pasientenes sannsynlighetsfordelinger til prognosegruppens fordelinger. Dette gjorde vi for å beholde et visst samsvar mellom bruken av adaptive egenskaper og fremgangsmåten

Tabell 5.35: Gjennomsnittlig andel forekomster av entropi-grupper, basert på pasientene fra treningsettet i M98.

Entropi-gr.:	1	2	3	4	5	6	7	8	9	10
Andel i %:	$< 10^{-2}$	$< 10^{-2}$	$< 10^{-3}$	< 0.1	$< 10^{-2}$	< 0.1	< 0.1	< 1	3	96

for bruk av utvalgte TU- og VAR_c -grupper, men dette er ikke nødvendigvis den beste fremgangsmåten når man jobber med adaptive egenskaper. I [45] og [49] antar Nielsen et al., i likhet med oss, at dataene er normalfordelte. Her bruker de imidlertid forventningsverdien og standardavviket til F_+ og F_- , og lar dette være egenskapene til hver pasient. Med denne fremgangsmåten oppnår de gode resultater, og det kan tenkes at vi kunne oppnådd bedre resultater ved å gjøre dette, fremfor å måle avstanden mellom sannsynlighetsfordelinger. Vi vil komme tilbake til dette når vi diskuterer tilsvarende fremgangsmåte som her, men benyttet på L23 i stedet for M98. Der vil vi se at vår bruk av de adaptive egenskapene ikke nødvendigvis er opphavet til problemet. Det kan derimot tenkes at et for lite datasett er grunnen til dårlige resultater etter test, slik vi har opplevd tidligere.

5.10.2 Kombinasjon mellom TU og varians

Som vi så i resultatene i seksjon 5.9.2, endte vi opp med heller dårlige resultater. Vi er forsåvidt vant med at test-resultatene blir dårlige i M98, men heller ikke resultatene etter trening ble særlig gode i dette tilfellet. Her endte vi opp med en CCReq på 60 % etter trening, mens i 5.1 så vi at en TU-gruppe oppnådde en CCReq på 80 % på samme treningsett. Vi endte også opp med en CCReq på 70 % etter trening i seksjon 5.9, hvor de adaptive egenskapene ble basert på kombinasjonen mellom TU og entropi. Dette passer også overens med det vi observerte fra korrelasjonen mellom god og dårlig prognose.

Når vi går gjennom resultatene etter test, ser vi også her at resultatet blir dårligere enn ved tilfellet hvor entropi ble brukt i stedet for VAR_c . Kombinasjonen med entropi endte med en CCReq på 29 % i test, mens TU kombinert med VAR_c resulterte i en CCReq på 25 %. Foreløpig ser det derfor ut til at TU kombinert med VAR_c utgjør en litt dårligere kombinasjon enn TU kombinert med entropi. Dette passer godt med det vi så i seksjon 4.4, hvor det så ut til at TU kombinert med entropi førte til litt mer spredning av cellekjerne fra pasienter med god prognose, enn cellekjerne fra pasienter med dårlig prognose. Det er imidlertid vanskelig å trekke gode konklusjoner ut fra resultatene i M98, ettersom dette datasettet består av såpass få pasienter. I tillegg har vi sett at dette lave antallet pasienter ikke medfører så troverdige resultater. Vi kommer derfor til å se videre på hvilken kombinasjon som kan gjøre det best, når vi tar for oss resultatene fra L23.

5.11 Resultater fra bruk av adaptive egenskaper benyttet på nabolag bestående av 8 piksler på originale bilder fra L23

I denne seksjonen blir resultatene av å anvende de adaptive egenskapene på de originale cellekjerne-bildene fra L23 presentert.

5.11.1 Kombinasjon av TU og entropi

Ved bruk av adaptive egenskaper som kombinerer TU og ENT , satt vi først opp en korrelasjonsmatrise for å se hvor like F_+ og F_- er. Dette gjør vi for å kunne avgjøre om vi skal benytte både F_+ og F_- , eller kun en av egenskapene. Korrelasjonsmatrisen ble

	F_+ -god	F_+ -dårlig	F_- -god	F_- -dårlig
F_+ -god	1.0	1.0	1.0	1.0
F_+ -dårlig		1.0	1.0	1.0
F_- -god			1.0	1.0
F_- -dårlig				1.0

Her er det verdt å merke seg at korrelasjonene er avrundet for bruk av kun en desimal, og at korrelasjonen mellom god og dårlig prognose ikke er nøyaktig 1. Som vi kan se fra matrisen er F_+ og F_- helt overlappende, og av den grunn velger vi å kun benytte F_+ . Vi kan også legge merke til at god og dårlig prognose er veldig korrelatert, noe som indikerer at vi ikke vil ende opp med særlig gode resultater. Som vi kan se fra tabell 5.36 ble resultatet fra trening 54 %. Det ser med andre ord ikke så lovende ut med tanke på testing. Når vi nå ser på tabell 5.37, ser vi at testresultatet ble 51 %. Dette er ikke et godt resultat. Diskusjon omkring disse resultatene er å finne i seksjon 5.12.

5.11.2 Kombinasjon av TU og varians

Ved bruk av adaptive egenskaper som kombinerer TU og VAR_c , satt vi først opp en korrelasjonsmatrise for å se hvor like F_+ og F_- er. Dette gjør vi for å kunne avgjøre om vi skal benytte både F_+ og F_- , eller kun en av egenskapene. Korrelasjonsmatrisen ble

	F_+ -god	F_+ -dårlig	F_- -god	F_- -dårlig
F_+ -god	1.0	0.9	1.0	0.9
F_+ -dårlig		1.0	0.9	1.0
F_- -god			1.0	0.9
F_- -dårlig				1.0

Tabell 5.36: Resultater etter klassifisering av treningsettet i L23. Resultatene kommer fra bruk av adaptive metoder med entropi.

F_+	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	46 %
Gruppe 0:	18	31	49	Spesifisitet:	37 %
Gruppe 1:	5	13	18	Sensitivitet:	72 %
Totalt:	23	44	67	CCReq:	54 %

Tabell 5.37: Resultater etter klassifisering av testsettet i L23. Resultatene kommer fra bruk av adaptive metoder med entropi.

F_+	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	48 %
Gruppe 0:	21	27	48	Spesifisitet:	44 %
Gruppe 1:	8	11	19	Sensitivitet:	58 %
Totalt:	29	38	67	CCReq:	51 %

Tabell 5.38: Resultater etter klassifisering av treningsettet i L23. Resultatene kommer fra bruk av adaptive metoder med varians.

F_+	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	66 %
Gruppe 0:	40	9	49	Spesifisitet:	82 %
Gruppe 1:	14	4	18	Sensitivitet:	22 %
Totalt:	54	13	67	CCReq:	52 %

Som vi kan se fra matrisen er F_+ og F_- helt overlappende, og av den grunn velger vi å kun benytte F_+ . I likhet med TU kombinert med ENT, ser vi også her at god og dårlig prognose er veldig korrelert, og nok en gang kan vi ikke forvente særlig gode resultater. I tabell 5.38 kan vi se at resultatet ble 52 % på trening. Det ser med andre ord ikke så lovende ut med tanke på testing. Når vi nå ser på tabell 5.39, ser vi at testresultatet ble 57 %. Dette er ikke et godt resultat, men det er en forbedring på 5 prosentpoeng fra trening. Mer diskusjon omkring disse resultatene er å finne i seksjon 5.12.

5.11.3 Kombinasjon av TU og gråtone

I likhet med tidligere setter vi først opp en korrelasjonsmatrise for å se hvor korrelerte F_+ og F_- er. Korrelasjonsmatrisen ble

	F_+ -god	F_+ -dårlig	F_- -god	F_- -dårlig
F_+ -god	1.0	1.0	1.0	1.0
F_+ -dårlig		1.0	1.0	1.0
F_- -god			1.0	1.0
F_- -dårlig				1.0

Som vi kan se fra matrisen er F_+ og F_- helt overlappende, og av den grunn velger vi å kun benytte F_+ . I likhet med de to andre kombinasjonene benyttet ved adaptive egenskaper på L23, ser vi også her at god og dårlig prognose er veldig korrelert, og nok en gang kan vi ikke forvente særlig gode resultater. Korrelasjonen mellom god og dårlig prognose er ikke nøyaktig 1, men er større enn 0.95. I tabell 5.40 kan vi se at resultatet, i form av CCReq, kom på 50 % på trening. Det ser med andre ord ikke så lovende ut med tanke på testing. Når vi nå ser på tabell 5.41, ser vi at CCReq etter test også kom på

Tabell 5.39: Resultater etter klassifisering av testsettet i L23. Resultatene kommer fra bruk av adaptive metoder med varians.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	40	8	48	Spesifisitet:	83 %
Gruppe 1:	13	6	19	Sensitivitet:	32 %
Totalt:	53	14	67	CCReq:	57 %

Tabell 5.40: Resultater etter klassifisering av treningsettet i L23. Resultatene kommer fra bruk av adaptive metoder basert på TU kombinert med senterpikslens gråtone.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	45 %
Gruppe 0:	19	30	49	Spesifisitet:	39 %
Gruppe 1:	7	11	18	Sensitivitet:	61 %
Totalt:	26	41	67	CCReq:	50 %

Tabell 5.41: Resultater etter klassifisering av testsettet i L23. Resultatene kommer fra bruk av adaptive metoder basert på TU kombinert med senterpikslens gråtone.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	49 %
Gruppe 0:	23	25	48	Spesifisitet: 48 %	
Gruppe 1:	9	10	19	Sensitivitet: 53 %	
Totalt:	32	35	67	CCReq: 50 %	

50 %. Dette er dårlige resultater og mer diskusjon omkring disse resultatene er å finne i seksjon 5.12.

5.12 Diskusjon rundt adaptive egenskaper benyttet på de originale bildene i L23

I denne seksjonen diskuteres bruken og resultatene av de adaptive egenskapene på de originale bildene fra L23. Vi starter med en diskusjon omkring kombinasjonen av TU og entropi, for deretter å diskutere kombinasjonen av TU og varians.

5.12.1 Kombinasjon mellom TU og entropi

Når vi skal se på resultatene vi kom frem til ved bruk av adaptive egenskaper basert på TU og entropi, er det rimelig å sammenlikne disse resultatene med resultatene vi så i seksjon 5.5.1. Her kan vi se at resultatene vi ender opp med ved bruk av adaptive egenskaper ligger midt mellom resultatene etter både trening og test for TU i seksjon 5.5.1. I spredningsplottene fra trening og test, som er å finne i figur 5.15, kan vi se at pasientene i de to prognosegruppene ligger forholdsvis spredt, og vi ser ingen tydelig gruppering av pasienter fra en prognose-gruppe.

Ettersom de adaptive egenskapene er vektet i forhold til hvilke deler av TU-Entropi-matrisen som inneholder størst avstander mellom de to prognosegruppene, forventet vi at klassifiseringsresultatet skulle bli minst like gode som de beste i seksjon 5.5.1. Slik gikk det imidlertid ikke. Ved bruk av adaptive egenskaper endte vi opp med en CCReq på 54 % etter trening, hvilket er 21 prosentpoeng dårligere enn beste tilsvarende resultat i 5.5.1 og 12 prosentpoeng dårligere enn det nest beste. Vi ser også at bruken av de adaptive egenskapene ikke medfører generalisering av resultatene, ettersom CCReq etter test kom på 51 %. I 5.5.1 så vi at det beste testresultatet kom på 63 %, noe som betyr at resultatet vi fikk med adaptive egenskaper ble 12 prosentpoeng dårligere. Da vi så på resultatene etter bruk av adaptive

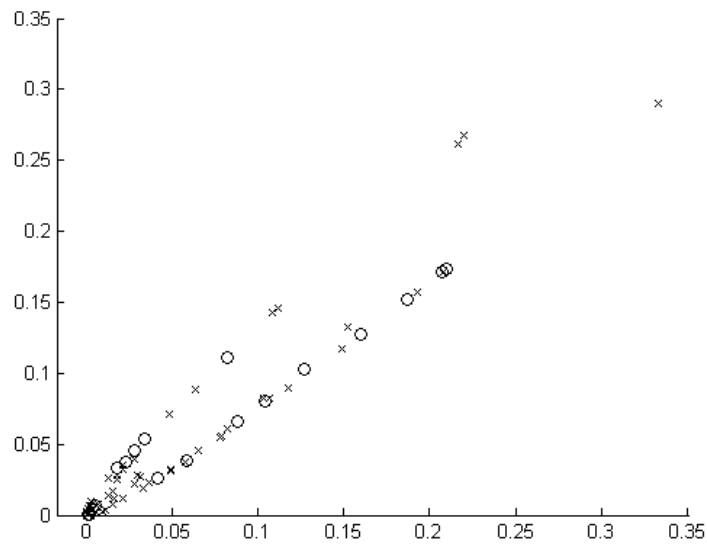
egenskaper med entropi, så vi at korrelasjonen mellom distansene til hver pasient med god og dårlig prognose var veldig høy. Av den grunn er klassifiseringsresultater nærmere 50 % forventet.

De adaptive egenskapene er basert på TU kombinert med entropi. Entropi valgte vi ettersom denne egenskapen tilsynelatende så ut til å gi mest skille mellom prognosegruppene. Vi baserte dette på treningsettet i M98, og det trenger ikke å gjelde for L23. For å se om entropi faktisk har en nytteverdi for å skille prognosegruppene i L23, går vi frem på samme måte som i seksjon 4.4.4. Spredningsplottet som viser gjennomsnittlig TU og entropi i cellekjerner er å finne i figur 5.16. Som vi kan se er det fortsatt mange cellekjerner fra pasienter med god prognose med gjennomsnittlig TU- og entropi-gruppe, som cellekjernene fra pasientene med dårlig prognose ikke har. Det tyder derfor på at TU og entropi fortsatt tilsynelatende utgjør en god kombinasjon for å skille de to prognoseklassene. Med andre ord holder fortsatt valget vi gjorde basert på M98 også for L23.

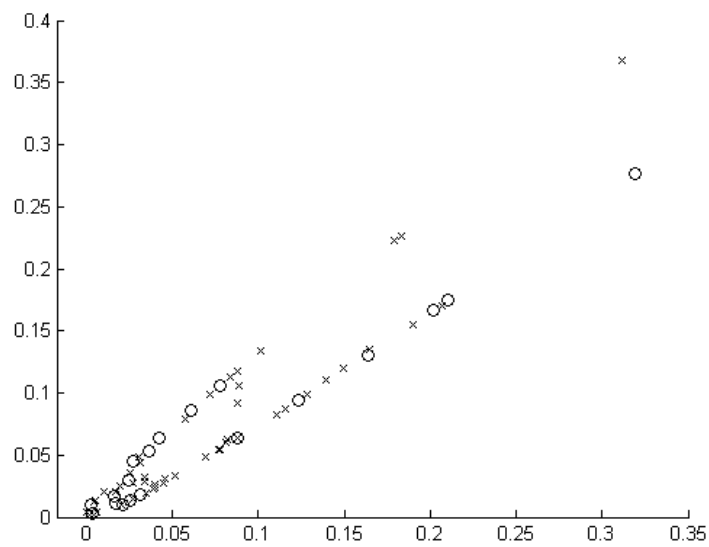
Vi har tidligere lurt på om bruken av Gaussisk Bhattacharyya som distanse mål fortsatt kan forsvares når man skal måle avstandene mellom sannsynlighetsfordelinger fra pasienter i L23. Tidligere har vi sett at dette kan forsvares, men vi vet ikke med sikkerhet om det kan forsvares her også. Av den grunn ser vi på sannsynlighetsfordelingene til de to prognosegruppene i både trening og test for F_+ . Disse er å finne i figur 5.17. Her observerer vi at antakelse om Gaussisk fordeling fortsatt holder. Vi kan også se at fordelingene basert på god prognose likner forholdsvis mye på hverandre i både forventningsverdi og varians, men dette gjelder ikke i like stor grad for fordelingene basert på dårlig prognose. Den største likheten mellom fordelingene basert på dårlig prognose er forventningsverdien, noe som betyr at større vektning av forventningsverdien fremfor standardavviket kan være veien å gå. Av den grunn kunne det lønt seg å bruke Mahalanobis fremfor Gaussisk Bhattacharyya i dette tilfellet. Vi skal imidlertid ikke legge for stor vekt på at dette er grunnen til de dårlige resultatene, ettersom Gaussisk Bhattacharyya også får med seg dette. I tillegg kan vi også se at begge fordelingene basert på pasientene med dårlig prognose har lavere standardavvik enn fordelingene basert på pasientene med god prognose. Mye tyder derfor på at fremgangsmåten vår etter at F_+ og F_- ble funnet har lite for seg.

De dårlige resultatene vi oppnådde med adaptive egenskaper, stemmer ikke overens med det Nielsen et al. erfarte i [45]. De benyttet imidlertid en litt annen fremgangsmåte etter at F_+ og F_- ble funnet, og det kan være nettopp dette som gjør at de ender opp med så mye bedre resultater enn oss. Vi var inne på dette da vi diskuterte resultatene fra bruk av adaptive egenskaper på M98. Der så vi imidlertid en stor likhet mellom resultatene fra adaptive egenskaper og tidligere resultater vi hadde oppnådd på M98. Det virket derfor som om de dårlige resultatene kom fra et datasett med få pasienter, fremfor vår bruk av F_+ og F_- . Ettersom de dårlige resultatene blir videreført til L23, valgte vi å se nærmere på om dette faktisk var tilfellet. Vi valgte derfor å se om bruk av F_+ og F_- på lik måte som i [45] ga bedre

resultater. Her fikk vi hjelp av Birgitte Nielsen til å hente ut egenskaper fra F_+ og F_- , samt utføre selve klassifiseringen. Dette resulterte i en CCR_{eq} på 50 % på trening og 46 % på test, for både F_+ og F_- . Disse resultatene er dårligere enn det vi oppnådde med vår fremgangsmåte, så det er ikke på grunn av fremgangsmåten vi benytter oss av etter at F_+ og F_- er beregnet, som er grunn til de dårlige resultatene. Det kan heller tyde på at den adaptive kombinasjonen av entropi og LBP ikke er en god fremgangsmåte. Dette tilsier at vår måte å beregne F_+ og F_- på er det svake leddet, til tross for at TU kombinert med entropi tilsynelatende medførte en del ulikheter mellom klassene.

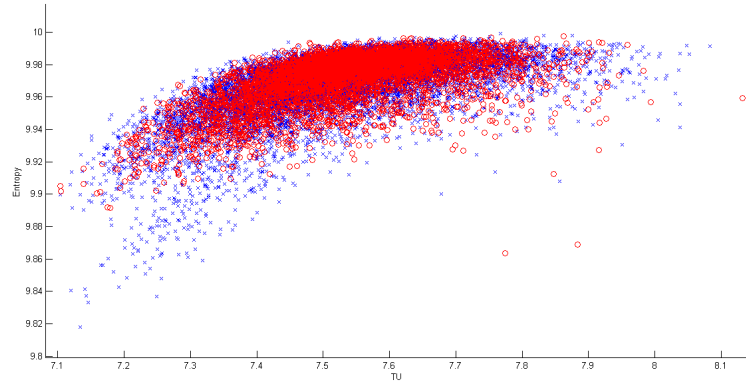


(a) Alle pasienter i treningssettet

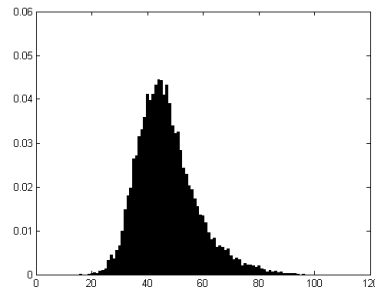


(b) Alle pasienter i testsettet

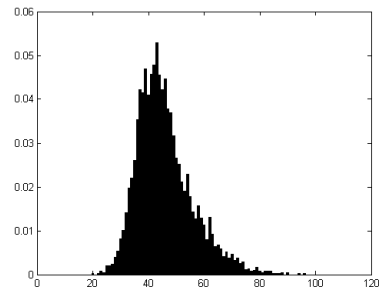
Figur 5.15: Denne figuren viser spredningsplottene basert på distanser til pasienter i trening- og testsettet i L23. X representerer god prognose, mens O representerer dårlig prognose. Distanse til klassen med god prognose er vist på x -aksen, mens y -aksen viser pasientens distanse til klassen med dårlig prognose.



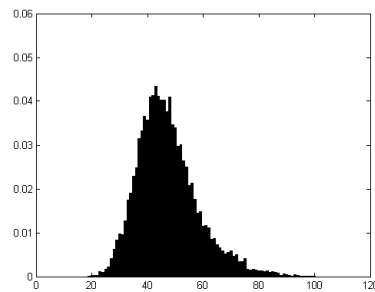
Figur 5.16: Figuren viser alle celler i treningsettet i L23 i form av deres gjennomsnittlige TU- og entropi-gruppe. Legg merke til at verdiene langs y-aksen representerer numrene vi så i tabell 4.4, men hvor den andre forekomsten av entropi lik 2 er sett bort ifra.



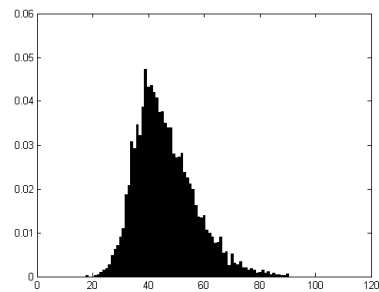
(a) God prognose - Trening.



(b) Dårlig prognose - Trening.



(c) God prognose - Test.



(d) Dårlig prognose - Test.

Figur 5.17: Denne figuren viser sannsynlighetsfordelinger til pasienter med god og dårlig prognose fra treningsettet i L23, hvor pasienter med god og dårlig prognose er hver for seg. Fordelingene er basert på F_+ .

5.12.2 Kombinasjon mellom TU og varians

Det første vi legger merke til når man ser på resultatene, er at CCR_{eq} etter trening ble veldig lav, og kom på kun 52 %. Allerede her ser vi at denne fremgangsmåten har lite for seg. Resultatene etter test ble litt bedre, med en CCR_{eq} på 57 %, men det er fortsatt ikke et bra resultat.

I seksjon 5.5 så vi at resultatene etter bruk av $LBP_{8,1}$ og $VAR_{c8,1}$ oppnådde CCR_{eq} på over 60 %. Det er bedre resultater enn det vi ender opp med her, og vi kan tydelig se at den adaptive fremgangsmåten som kombinerer TU- og VAR_c -grupper ikke gjør det like godt. Spørsmålet som gjenstår er hvorfor denne adaptive fremgangsmåten feiler.

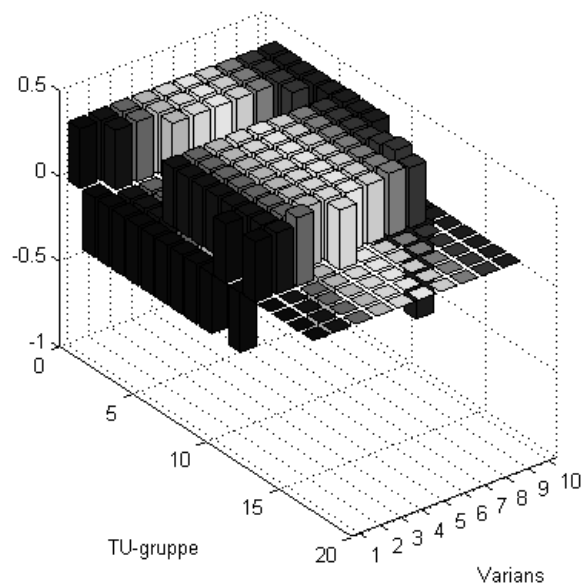
For å svare på hvorfor den adaptive fremgangsmåten ikke gjør det bra, ser vi på hvilke TU/ VAR_c -kombinasjoner som har de høyeste distansene. Dette kan vi observere i figur 5.18. I likhet med tidligere blir ikke kombinasjoner gitt vektor, hvis kombinasjonen ikke forekommer minst en gang i hver av de to prognosegruppene. Vi kan med andre ord se at nesten alle TU/ VAR_c -kombinasjoner blir vektet likt ved utregningen av F_+ og F_- . Ettersom vi kun ser på F_+ , er det de positive distansene som er av interesse for oss, når vi skal se hva som gjør at denne fremgangsmåten ikke leverer så gode resultater som vi hadde håpet på. Her kan vi se at det er TU-gruppene 1-3 og 6-11, samt alle VAR_c -grupper, med unntak av VAR_c -gruppe 1 og delvis 10, som ender opp med positive distanser. Ettersom alle disse distansene er like, er det kun andelen forekomster av disse TU- og VAR_c -gruppene som avgjør hva F_+ blir beregnet til.

Vi husker fra tidligere at vi kvantiserte alle VAR_c -verdiene slik at hver VAR_c -gruppe inneholdt tilnærmet like mange forekomster. Av den grunn vil alle VAR_c -grupper mellom 2 og 9 ha tilnærmet like mye å si når F_+ blir beregnet. Vi vet fra 5.5 at ikke alle VAR_c -gruppene oppnådde like gode resultater. Av den grunn mister vi litt klasseskillende informasjon når både de ”gode” og ”dårlige” VAR_c -gruppene har like mye å si.

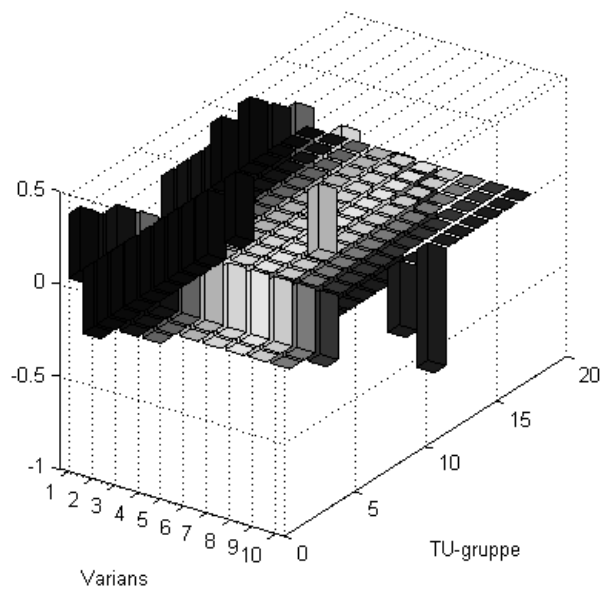
Dersom vi går over til å se på forekomstene av TU, som er illustrert i figur 5.19, kan vi se at det er TU-gruppe 7, 8 og 9 som dominerer. Fra seksjon 5.5.1 husker vi at det kun var TU-gruppe 8, av disse tre, som oppnådde gode resultater etter trening. Derimot var det ingen av disse som gjorde det bra i test. Med andre ord blir F_+ i stor grad dominert av forekomster som tilsynelatende har lite klasseskillende informasjon. I likhet med det vi så i seksjon 5.12, ser vi også her at det er grunnlaget til F_+ , som fører til at vi ender opp med såpass dårlige klassifiseringsresultater.

5.12.3 Kombinasjon mellom TU og senterpikselens gråtone

I likhet med bruken av adaptive egenskaper basert på TU og entropi, samt TU og VAR_c , observerte vi i seksjon 5.11.3 at heller ikke TU kombinert med senterpikselen ga gode resultater. Ettersom CCR_{eq} etter trening og test ble

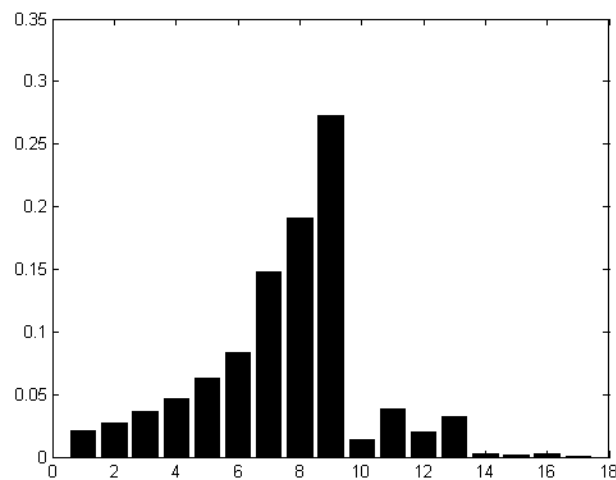


(a) Denne figuren viser distansene med hensyn på den positive delen.



(b) Denne figuren viser distansene med hensyn på den negative delen.

Figur 5.18: Figuren viser distansene mellom de to prognosegruppene for hver kombinasjon av TU og VAR_c . Distansene er funnet på grunnlag av pasientene i treningsettet i L23.



Figur 5.19: Figuren viser andelen forekomster av de ulike TU-gruppene i L23.

50 %, kan vi konkludere med at fremgangsmåten gir lite klasseskillende informasjon.

Ettersom både entropi og VAR_c fikk dårlige resultater på grunn av hvordan de adaptive egenskapene ble beregnet, er det høyst sannsynlig også tilfellet for bruken av senterpikselens gråtone. Vi kommer ikke til å gå dypere inn på hva som gjør at resultatene blir så dårlige, ettersom vi allerede har en god formening om hvor problemet ligger.

5.13 Resultater fra bruk av adaptive egenskaper benyttet på nabolag bestående av 8 piksler på nedskalerte bilder fra L23

Vi skal nå gå gjennom resultatene fra bruken av de adaptive egenskapene, med 8 piksels nabolag, på de nedskalerte cellekjerne-bildene fra L23.

5.13.1 Kombinasjon av TU og entropi

Ved bruk av adaptive egenskaper som kombinerer TU og ENT , satt vi først opp en korrelasjonsmatrise for å se hvor like F_+ og F_- er. Dette gjør vi for å kunne avgjøre om vi skal benytte både F_+ og F_- , eller kun én av egenskapene. Korrelasjonsmatrisen ble

	F_+ -god	F_+ -dårlig	F_- -god	F_- -dårlig
F_+ -god	1.0	0.9	1.0	0.9
F_+ -dårlig		1.0	0.9	1.0
F_- -god			1.0	0.9
F_- -dårlig				1.0

Som vi kan se fra matrisen er F_+ og F_- helt overlappende, og av den grunn velger vi å kun benytte F_+ . I likhet med bruk av adaptive egenskaper basert på TU og entropi, er også pasientene fra de to prognosegruppene svært korrelerte. Av den grunn kan vi forvente dårlige resultater. På trening ble det oppnådd CCReq på 50 %, se tabell 5.42. Dette er dårlige resultater, så vi kan ikke forvente godt testresultat. I tabell 5.43 kan man se at resultatet på test ble 55 %, noe som er bedre enn resultatet fra trening. For nærmere diskusjon se seksjon 5.12.

5.13.2 Kombinasjon av TU og varians

Til tross for dårlige resultater etter bruk av adaptive egenskaper basert på TU kombinert med entropi, velger vi å benytte adaptive egenskaper basert på kombinasjon mellom TU og VAR_c ettersom denne kombinasjonen er lettest å sammenlikne med kombinasjonene vi utførte med klassiske egenskaper. På samme måte som tidligere har vi satt opp en korrelasjonsmatrise mellom F_+ og F_- , og denne ble

	F_+ -god	F_+ -dårlig	F_- -god	F_- -dårlig
F_+ -god	1.0	0.6	1.0	0.6
F_+ -dårlig		1.0	0.6	1.0
F_- -god			1.0	0.6
F_- -dårlig				1.0

Tabell 5.42: Resultater etter klassifisering av nedskalerte cellekjerne-bilder fra trening i L23, ved benyttelse av adaptive egenskaper.

F_+	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	46	3	49	Spesifisitet:	94 %
Gruppe 1:	17	1	18	Sensitivitet:	6 %
Totalt:	63	4	67	CCReq:	50 %

Tabell 5.43: Resultater etter klassifisering av nedskalerte cellekjerne-bilder fra test i L23, ved benyttelse av adaptive egenskaper.

F_+	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	67 %
Gruppe 0:	44	4	48	Spesifisitet:	92 %
Gruppe 1:	18	1	19	Sensitivitet:	5 %
Totalt:	62	5	67	CCReq:	48 %

Tabell 5.44: Resultater etter klassifisering av treningssettet i L23. Resultatene kommer fra bruk av adaptive metoder med varians brukt på de nedskalerte bildene.

F_+	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	23	26	49	Spesifisitet:	47 %
Gruppe 1:	4	14	18	Sensitivitet:	78 %
Totalt:	27	40	67	CCReq:	62 %

Som vi kan se, er korrelasjonskoeffisienten mellom F_+ og F_- 1 i alle tilfeller, og av den grunn velger vi å kun bruke F_+ videre. Fra tabell 5.44 kan vi se at CCReq på trening ble 62 %, mens i test falt CCReq ned til 54 %, se tabell 5.45.

5.13.3 Kombinasjon av TU og senterpikselens gråtone

Ettersom resultatene etter bruk av adaptive egenskaper basert på TU og entropi, samt TU og VAR_c på nedskalerte cellekjerne-bilder i L23 og resultatene etter bruk av adaptive egenskaper basert på TU og senterpikselens gråtone på cellekjerne-bildene i L23 ble dårlige, ser vi ingen grunn til å benytte adaptive egenskaper basert på TU og senterpikselens gråtone på nedskalerte cellekjerne-bilder i L23.

Tabell 5.45: Resultater etter klassifisering av testsettet i L23. Resultatene kommer fra bruk av adaptive metoder med varians brukt på de nedskalerte bildene.

F_+	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	51 %
Gruppe 0:	22	26	48	Spesifisitet:	46 %
Gruppe 1:	7	12	19	Sensitivitet:	63 %
Totalt:	29	38	67	CCReq:	54 %

5.14 Diskusjon rundt adaptive egenskaper benyttet på nedskalerte bilder i L23

I denne seksjonen diskuterer vi bruken og resultatene av de adaptive egenskapene benyttet på de nedskalerte bildene fra L23. Vi starter med å diskutere kombinasjonen av TU og entropi, for deretter å diskutere kombinasjonen av TU og varians.

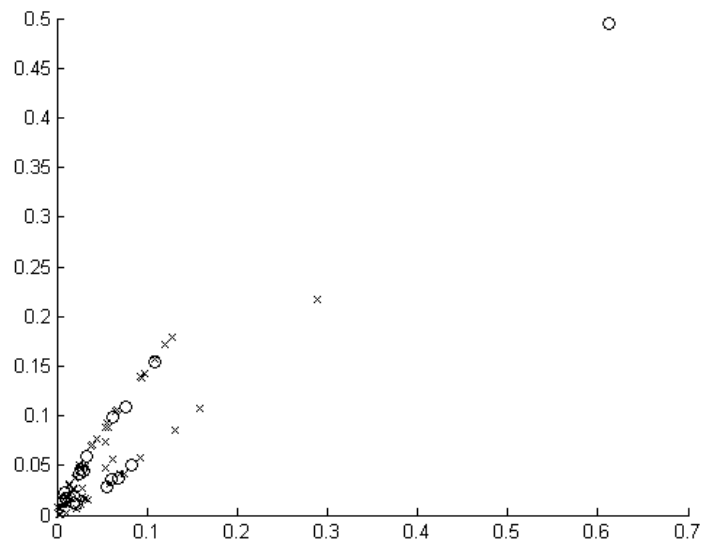
5.14.1 Kombinasjon mellom TU og entropi

Vi husker fra seksjon 5.7 at samtlige resultater, etter en nedskalering av cellekjerne-bildene før $LBP_{8,1}$ og $VAR_{c8,1}$ ble benyttet på bildene, var svært gode. Man skulle tro at det samme ville skje etter bruk av adaptive egenskaper, men så heldige var vi ikke. Her endte vi opp med en CCR_{eq} på 50 % etter trening og 48 % etter test. Dette var ikke uventede resultater, med tanke på korrelasjonen mellom de to prognosegruppene i treningsettet. Det er svært dårlige resultater, og i figur 5.20 kan vi se at distansene som blir brukt i klassifiseringen ikke bidrar til å skille pasienter i ulike prognosegrupper fra hverandre.

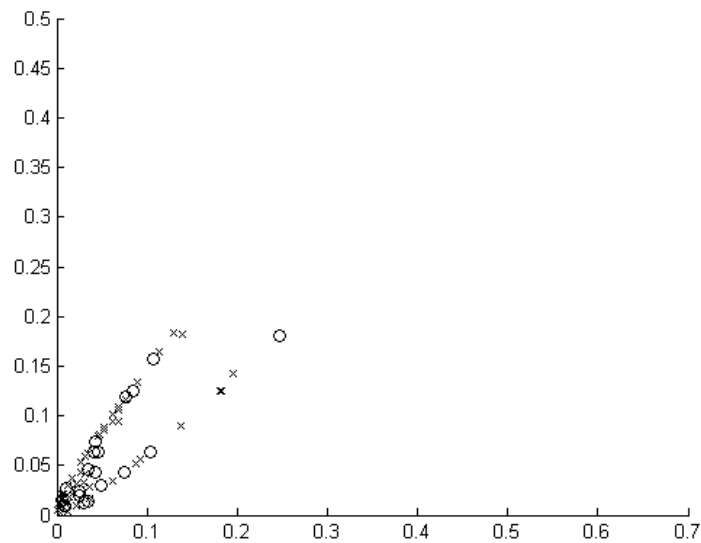
I spredningsplottet basert på distansene i treningsettet kan vi se at en pasient med dårlig prognose har veldig høye distanser i forhold til de resterende pasientene. Ettersom det kun er 18 pasienter med dårlig prognose i treningsettet, vil denne pasienten påvirke klassifikatoren i forholdsvis stor grad. Hadde vi sett bort fra denne pasienten kan det tenkes at vi kunne endt opp med bedre klassifiseringsresultater.

Ettersom vi jobber med adaptive metoder basert på hvor ulikhetene er størst i de to prognoseklassene, er det ikke utenkelig at den ene pasienten med forholdsvis høy distanse i treningsettet, fører til at kombinasjonen av TU og entropi blir vektet feil. Selv om dette ikke nødvendigvis er den eneste forklaringen på de dårlige resultatene kan vi i det minste se at dette er en av faktorene. Hvis vi ser bort fra den ene pasienten i sannsynlighetsfordelingen fra trening, observerer vi at spredningsplottene for trening og test blir mer like hverandre. I figur 5.21 kan man se spredningsplottene basert på treningsettet uten pasienten med relativt høye avstander. Ser vi bort den ene "ekstreme" pasienten ender vi opp med en CCR_{eq} på 69 % i trening, men dette faller igjennom i test, hvor CCR_{eq} kom på 57 %. Med andre ord ser det ut til at datasettet er litt lite, hvilket medfører høy påvirkning av uteliggere.

Ved bruk av adaptive metoder på L23 uten nedskalering endte vi, i likhet med her, opp med dårlige resultater, men der var det ikke en pasient som skilte seg så voldsomt ut som den ene vi til nå har skrevet om. Det tyder derfor på at vi ikke kan legge all skylden på en enkelt pasient, men at det er flere elementer som spiller inn i klassifiseringen. Vi var inne på flere ulike faktorer som kan påvirke resultatet i seksjon 5.12. Vi kommer ikke til å gjenta dette her, men det ser ut til at fremgangsmåten vi benytter for å

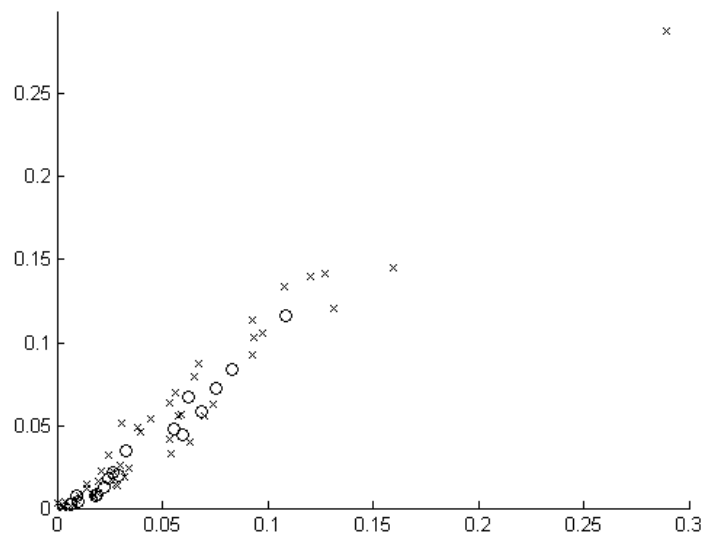


(a) Alle pasienter i treningssettet

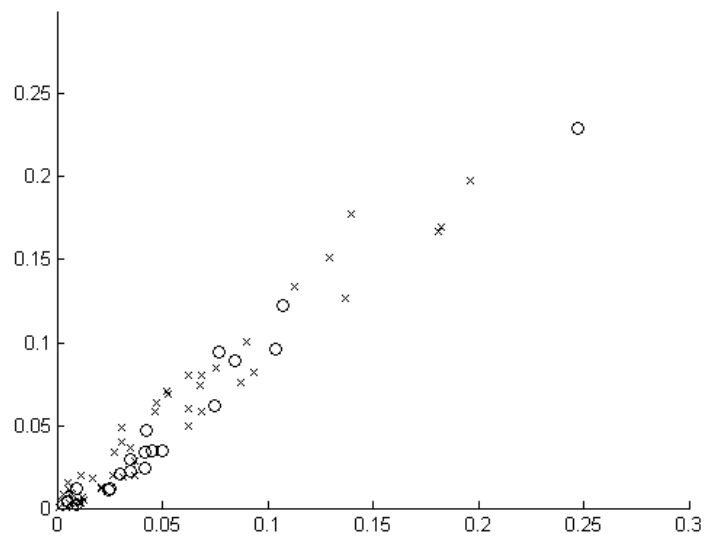


(b) Alle pasienter i testsettet

Figur 5.20: Denne figuren viser spredningsplottene basert på distanser til pasienter i trening- og testsettet i L23. Cellekjerne-bildene er nedskalert med en faktor på 2 før informasjon er hentet fra cellekjernene. X representerer god prognose, mens O representerer dårlig prognose. Distanse til klassen med god prognose er vist på x -aksen, mens y -aksen viser pasientens distanse til klassen med dårlig prognose.

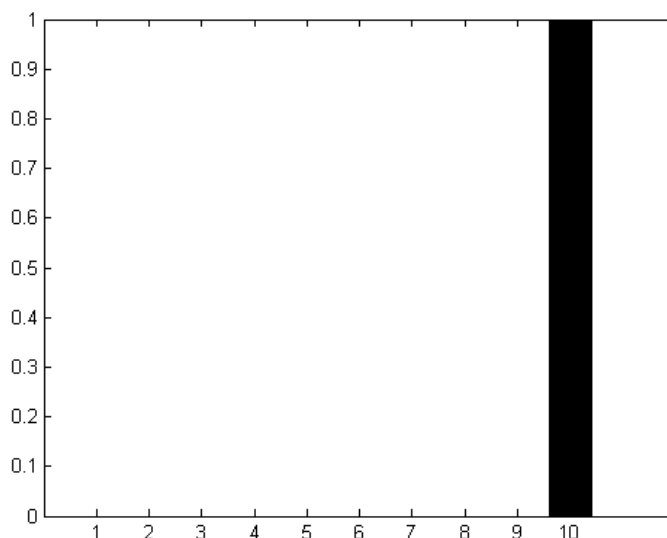


(a) Trening



(b) Test

Figur 5.21: Denne figuren viser spredningsplottene basert på distanser til pasienter i treningsettet i L23 hvor cellekjerne-bildene er nedskalert med en faktor på 2 før informasjon er hentet fra cellekjernene. Pasienten med veldig høye avstander i 5.20a er ikke blitt brukt i beregningene.



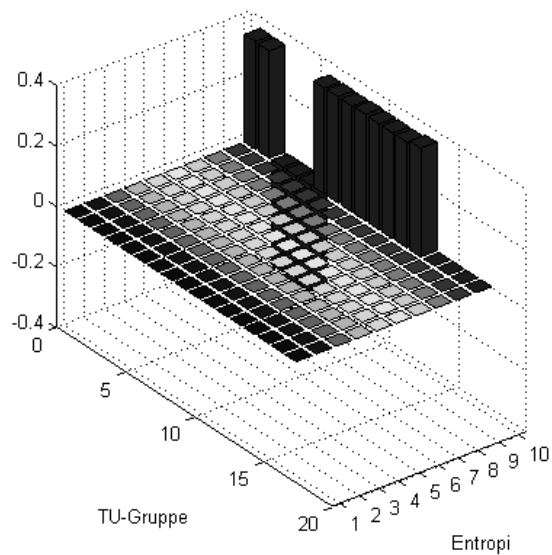
Figur 5.22: Denne figuren viser andelen forekomster av de ulike entropiene over alle nedskalerte cellekjerne-bilder i L23. Forekomstene er sortert på lik måte som i tabell 4.5.

finne F_+ er grunnen til de dårlige resultatene.

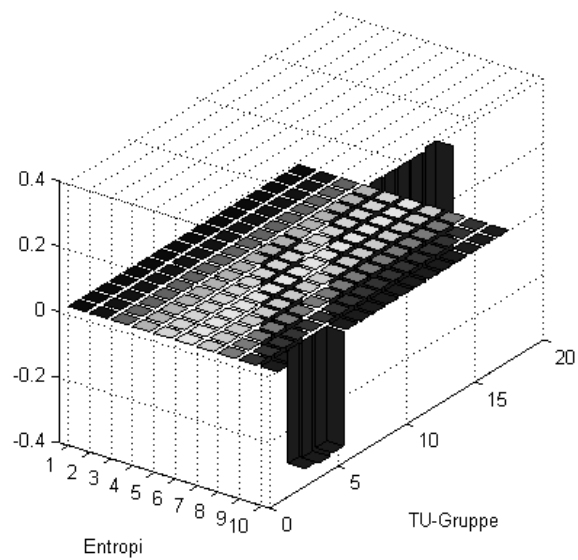
Dersom vi ser på figur 5.22 kan vi se at det kun er en entropi-verdi som forekommer. Vi kan med andre ord se at kombinasjon av TU og entropi har lite for seg, og at det ville vært bedre å se på TU alene. Dette gjenspeiles i figur 5.23 hvor vi kun ser distanser hos den høyeste entropi-gruppen. Alle ikke-null distanser er tilsynelatende like, noe som gjør at TU-gruppene 1, 2 og 6-13 blir vektet likt. Vi vet fra seksjon 5.7 at TU-gruppe 2, 6, 7 og 8 oppnådde gode resultater både på trening og test. Ettersom vi ikke oppnår gode resultater her, tyder det på at den klasseskillende informasjonen fra disse gruppene blir borte når flere dårlige TU-grupper blir vektet like mye. Det er verdt å merke seg at selv om vi hadde brukt full oppløsning av entropi-verdiene, ville ikke resultatet blitt anderledes. Dette kommer av at entropi-verdi 3 forekommer mer enn 99.9 % av alle gangene.

5.14.2 Kombinasjon mellom TU og varians

Vi husker fra seksjon 5.13.2 at resultatene etter trening var relativt gode i forhold til resultatene vi tidligere har sett på etter bruk av adaptive egenskaper. Her kom CCReq på 62 %, noe som gjør denne bruken av de adaptive egenskapene best på L23. En CCReq på 62 % etter trening er imidlertid ikke et godt resultat, spesielt med tanke på at vi allerede oppnår CCReq på over 70 % i flere tilfeller ved bruk av $LBP_{8,1}$ og $VAR_{c8,1}$ på nedskalerte cellekjerne-bilder i L23, noe vi så i seksjon 5.7. Ettersom testresultatene i 5.7 også kom på over 70 %, vet vi allerede nå at bruk av adaptive egenskaper basert på kombinasjon mellom TU og VAR_c blir utkonkurrert.



(a) Viser distansene med hensyn på den positive delen.



(b) Viser distansene med hensyn på den negative delen.

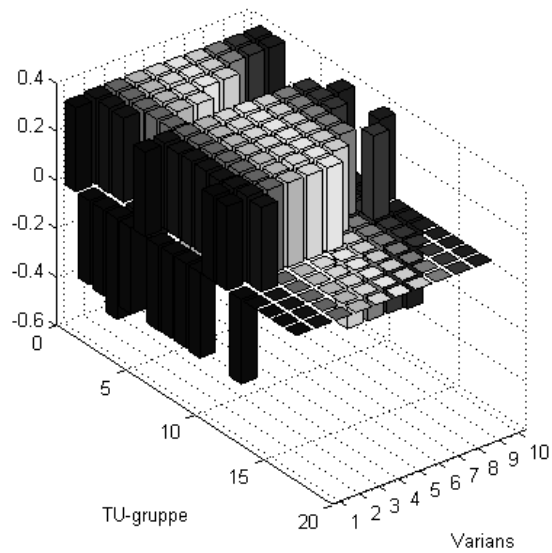
Figur 5.23: Figuren viser distansene mellom de to prognosegruppene for hver kombinasjon av TU og entropi. Distansene er funnet på grunnlag av pasientene i treningssettet i L23.

I likhet med tidligere ender vi ikke opp med et godt testresultat. Som vi så i seksjon 5.13.2 endte vi opp med en CCReq på 53 %, noe som betyr at denne metoden ikke klarer å skille prognosegruppene fra hverandre. Dette skyldes med høy sannsynlighet at F_+ blir dannet på et veldig dårlig grunnlag, slik vi har skrevet tidligere. Vi kommer uansett til å se nærmere på distansene som brukes som vektorer for å danne F_+ og F_- , for å bli sikre på at dette fremdeles er tilfellet.

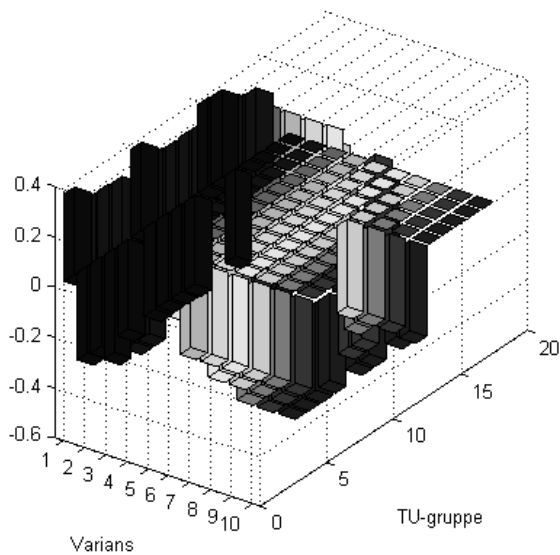
I figur 5.24 kan man se distansene mellom prognosegruppene, for hver kombinasjon av TU- og VAR_c -grupper, i form av et 2D-histogram. Figuren viser 2D-histogrammet med hensyn til både de positive og negative distansene. Her observerer vi at disse distansene er veldig like de vi har sett i tidligere seksjoner som også har omhandlet adaptive egenskaper. Ser man bort fra distansene som er lik 0, kan man se at de resterende distansene har forholdsvis lik absoluttverdi. Som vi skrev innledningsvis i seksjon 5.13.2 var F_+ og F_- svært korrelerte. Vi valgte derfor kun å se på F_+ , noe vi fortsetter med her. I figuren over distansene kan vi se at det er TU-gruppene 1-3 og 6-12 og VAR_c -gruppene 2-7 som er dominerende, men også tilnærmet like, og nok en gang er det kun andelen forekomster av de ulike TU- og VAR_c -gruppene som avgjør hva F_+ blir beregnet til.

Andelen forekomster i hver VAR_c -gruppe er tilnærmet like, ettersom kvantiseringen vi benyttet for $VAR_{c8,1}$ på nedskalerte cellekjerne-bilder i L23 ble dannet på dette grunnlaget. F_+ vil derfor være dannet med hensyn på VAR_c -gruppe 2-7, i omtrent like stor grad. Vi husker fra 5.7.2 at det kun var to VAR_c -grupper som oppnådde CCReq på over 60 % i både trening og test blant disse 6 gruppene. Dette var VAR_c -gruppe 6 og 7. I tillegg kan vi se at samtlige av disse 6 gruppene oppnådde greie resultater i trening, men falt igjennom i test, noe vi også kan se ved bruk av de adaptive egenskapene. Det er verdt å legge merke til at VAR_c -gruppe 1 og 8, som både gjorde det bra i trening og test, i seksjon 5.7.2, ikke er like dominerende i beregningen av F_+ . Vi kan derfor se at F_+ hverken blir dannet av kun de gode VAR_c -gruppene, eller alle de beste VAR_c -gruppene.

I figur 5.25 ser vi at det er TU-gruppe 2 og 9 som forekommer oftest i de nedskalerte cellekjerne-bildene i L23, og utgjør omtrent 50 % av alle forekomstene. Dette viser at beregningen av F_+ i stor grad blir dominert av disse to TU-gruppene. I seksjon 5.7 så vi at TU-gruppe 2 gjorde det bra i både trening og test, med CCReq på over 70 % i begge tilfellene, mens TU-gruppe 9 gjorde det litt dårligere med en CCReq mellom 60 og 70 % i både trening og test. Tar vi med TU-gruppe 8, som utgjør ca 15 % av alle TU-forekomstene, og som også gjorde det svært bra i både trening og test i 5.7, skulle man trodd at F_+ ble en forholdsvis god egenskap. Som vi var inne på tidligere ser det imidlertid ut til at kombinasjonen med VAR_c ødelegger, ettersom F_+ blir dannet av både gode og dårlige VAR_c -grupper.

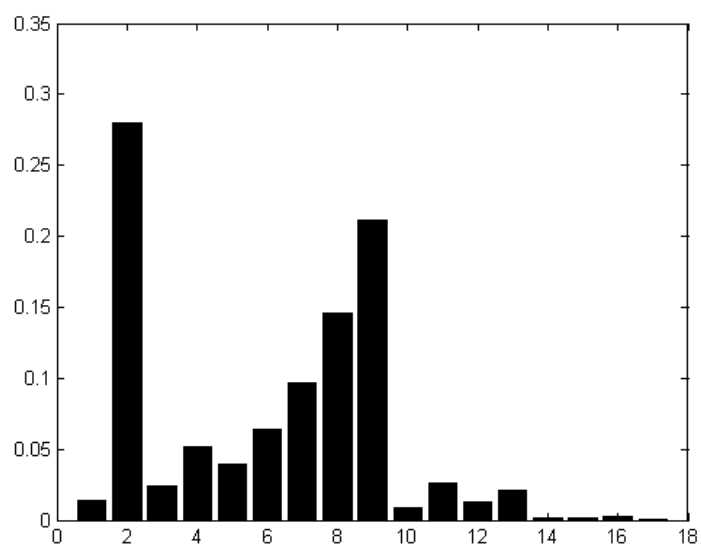


(a) Viser distansene med hensyn på den positive delen.



(b) Viser distansene med hensyn på den negative delen.

Figur 5.24: Figuren viser distansene mellom de to prognosegruppene for hver kombinasjon av TU- og VAR_c -grupper. Distansene er funnet på grunnlag av pasientene i treningsettet til de nedskalerte bildene i L23.



Figur 5.25: Figuren viser andelen forekomster av TU-grupper i de nedskalerte cellekjerne-bildene til pasientene i L23 sitt treningsettet.

5.15 Oppsummering rundt bruken av adaptive egenskaper

Fremgangsmåten vi benyttet ved bruk av adaptive egenskaper resulterte i høy korrelasjon mellom de to prognosegruppene. Dette ga svært dårlige resultater i alle forsøkene, og ble i samtlige tilfeller utkonkurrert av fremgangsmåtene hvor sannsynlighetsfordelingene til pasientene ble basert på forekomster av TU- og VAR_c -grupper.

Resultatene etter adaptive egenskaper basert på TU kombinert med VAR_c er litt bedre enn resultatene etter adaptive egenskaper basert på TU kombinert med entropi og TU kombinert med senterpikselens gråtone. Et unntak er resultatene fra M98, men som vi har vært inne på tidligere er M98 et veldig lite datasett. Det er av den grunn vanskelig å trekke konklusjoner fra M98 sammenliknet med L23. TU kombinert med VAR_c ga også bedre resultater enn TU kombinert med senterpikselens gråtone, som kun ble benyttet på originalbildene i L23.

Det kommer tydelig frem i diskusjonene som omhandler fremgangsmåtene hvor adaptive egenskaper har blitt brukt, at grunnen til de dårlige resultatene skyldes måten F_+ og F_- blir beregnet. I motsetning til oss ender Nielsen og Danielsen [50] opp med gode resultater på datasettet L23, med helt lik oppdeling som oss (med unntak av 3 pasienter som har skiftet prognosegruppe), ved benyttelse av adaptive egenskaper funnet fra GLCM, GLRLM, CGLRLM og CC. Det kan derfor se ut til at vi ville vært tjent med å benytte en annen fremgangsmåte for å beregne F_+ og F_- .

I tabell 5.35 observerte vi at entropi-gruppe 10 dominerte i cellekjernebildene. Det samme så vi i figuren 5.23. Denne entropi-gruppen består av entropi-verdiene 3 og 2.75, som forekommer når det er 1 eller 0 like gråtoner i 8-piksels nabolaget. Vi ender med andre ord opp med en veldig liten spredning av entropi-verdiene. Dersom vi hadde benyttet et større vindu kunne denne spredningen økt, og F_+ og F_- blitt beregnet på grunnlag av flere entropi-grupper. Da vi jobbet med adaptive egenskaper basert på VAR_c kombinert med TU, hjalp det imidlertid ikke at samtlige VAR_c -grupper forekom like ofte. Selv om det tilsynelatende var ulikheter mellom prognoseklassene da vi målte avstander mellom andelen TU- VAR_c -forekomster, kom ikke dette til nytte senere. Ved bruk av begge kombinasjonene erfarte vi at nesten alle kombinasjonene endte opp med omtrent lik distanse. Dette skjedde til tross for at vi vet at enkelte TU og VAR_c -grupper skiller klassene bedre enn andre. Det medfører at både gode og dårlige grupper blir vektet like mye, og av den grunn forsvinner den klasseskillende informasjonen enkelte grupper innehar.

Ettersom de adaptive egenskapene blir vektet på basis av cellekjernene til de to prognosegruppene, og ikke i forhold til hver enkelt pasients cellekjerne, kan det se ut til at den klasseskillende informasjonen ikke blir tatt med

i vektene. Selv om Nielsen og Danielsen gjør tilsvarende i [50], ender ikke de opp med dårlige resultater. Det tyder på at våre forekomstmatriser ikke egner seg godt for beregning av adaptive egenskaper slik som GLCM og CGLRLM.

Kapittel 6

Konklusjon og videre arbeid

Målet med denne oppgaven var å benytte statistisk teksturanalyse for å skille pasienter med god og dårlig kreftprognose. Vi har benyttet oss av materiale fra pasienter med prostatakreft (M98) og eggstokkreft stadium I (L23). Til teksturanalysen har vi benyttet Lokalt Binært Mønster, "Local Binary Pattern" (LBP) og lokal varians delt på senterpikselens gråtone, VAR_c , med 8 og 16 piksels nabolag på de originale bildene, samt 8 piksels nabolag på nedskalerte bilder. I tillegg har vi kombinert LBP og VAR_c , og jobbet med forskjellige kombinasjoner av LBP benyttet på ulike andeler av cellekjernen. Til slutt så vi på bruk av adaptive egenskaper, hvor vi kombinerte LBP med entropi, varians og senterpikselens gråtone.

Da vi jobbet med prostata-materialet (M98) erfarte vi at vi hadde et datasett med for få pasienter (20 pasienter i treningsettet og 14 pasienter i testsettet). Klassifiseringsresultatene snudde helt på test i forhold til trening. Da vi derimot så på eggstokk-materialet, som inneholder flere pasienter, erfarte vi ikke lenger dette. Vi har på grunnlag av dette erfart viktigheten av at treningsettet inneholder tilstrekkelig med pasienter. Resultatene etter våre forsøk på M98 endte med lave CCReq, men vi så at ved å gjøre dette datasettet større ble resultatene forholdsvis like de vi så for L23.

Vi har sett at de beste resultatene blir oppnådd ved bruk av $LBP_{8,1}$ og $VAR_{c8,1}$ på nedskalerte cellekjerne-bilder i L23. Her observerte vi at flere av de TU- og VAR_c -gruppene som gjorde det godt i test, ikke nødvendigvis gjorde det like godt i trening. Dette kommer av tilfeldigheter, og større datasett eller eventuelt egne valideringsett kunne gitt oss en bedre indikasjon på hvilken TU/ VAR_c -gruppe som oppnådde best resultater. For å bestemme det beste resultatet, velger vi for en gitt TU- eller VAR_c -gruppe å gå ut i fra det resultatet som er dårligst av trening og test. På denne måten endte vi opp med at TU-gruppe 7 gjorde det best, med CCReq på 77 % etter trening og 74 % etter test. Nielsen et al. har jobbet med nesten helt lik inndeling av pasientene i L23 som oss, og som vi allerede har sagt er den eneste forskjellen tre pasienter som i senere tid har blitt flyttet til motsatt prognosegruppe. I deres arbeid med denne oppdelingen av L23 oppnådde de et beste resultat på 70 % [43]. Vi ser dermed at vår fremgangsmåte gjør det

svært bra, og kan konkurrere med blant annet adaptive egenskaper hentet fra "Gray Level Cooccurrence Matrices" (GLCM), "Gray Level Run Length Matrices" (GLRLM), "Cooccurrence of Gray Level Run Length Matrices" (CGLRLM) og "Complexity Curves" (CC) [50].

Ved kombinerings mellom LBP og varians har vi ikke erfart de gode resultatene Ojala et al. tidligere har omtalt. I de beste tilfellene erfarte vi at kombinasjonen medførte noen prosents økning i andel suksessfull klassifisering, men ingen kombinasjoner førte til bedre resultater enn det den beste TU-gruppen oppnådde alene i nedskalerte cellekjerne-bilder. Det at slike kombinerings ikke var noen suksess skyldes ikke at VAR_c utgjorde en dårlig operator. I flere tilfeller erfarte vi at VAR_c oppnådde bortimot like gode resultater som LBP .

Da vi så på adaptive egenskaper merket vi oss at måten de adaptive egenskapene F_+ og F_- ble beregnet på ga opphav til dårlige egenskaper. Dette førte til dårlige resultater i samtlige forsøk. I alle forsøkene så vi at distansene til hver pasient basert på F_+ var svært korrelert til distansene basert på F_- . Av den grunn valgte vi å kun jobbe med F_+ . Videre erfarte vi at distansene fra pasientene med god prognose ble svært korrelert til pasientene med dårlig prognose, noe som medførte CCReq liggende rundt 50 % i både trening og test.

I denne oppgaven har vi vært igjennom en del ulike fremgangsmåter, men to fremgangsmåter skiller seg positivt ut. Disse går ut på å nedskalere cellekjerne-bildene med en faktor på to i hver retning, før enten $LBP_{8,1}$ eller $VAR_{c8,1}$ blir benyttet på disse. $LBP_{16,2}$ oppnådde ikke like gode resultater som $LBP_{8,1}$ benyttet på nedskalerte cellekjerne-bilder. Det ser likevel ut til at man ved å benytte LBP på cellekjerne-bilder er tjent med å se på en større andel av cellekjernen, enn det $LBP_{8,1}$ gjør når den er benyttet på originalbildene, hvor oppløsningen er 162 nm (M98) og 166 nm (L23) per piksel.

Videre arbeid

Selv om vi oppnådde dårlige resultater ved bruk av adaptive egenskaper, er det andre fremgangsmåter for beregning av adaptive egenskaper som kan gi bedre resultater. I stedet for å bruke et distansemål som grunnlag for vektfunksjonen for de adaptive egenskapene, kan man benytte en diskrimineringsmatrise [64]. Walker et al. oppnådde en korrekt klassifiseringsrate på 92 % [64] ved bruk av en diskrimineringsmatrise og GLCM som grunnlag for adaptive egenskaper.

I denne oppgaven har vi brukt sannsynlighetsfordelinger til å representere pasienter og pasientgrupper, og brukt distansemål basert på forventningsverdi og varians mellom gruppe- og pasientfordelinger. Vi observerte at det var forskjell mellom fordelingene for god og dårlig prognose, så det er tydelig

at det er klaseskillende informasjon å hente fra disse fordelingene. I flere tilfeller har det sett ut til at standardavviket i sannsynlighetsfordelingene har vært ulikt mellom god og dårlig prognose, og at dette synes i ytterkantene av fordelingene. Det kan derfor være verdt å gjøre et eksperiment hvor man benytter informasjonen som ligger i for eksempel den 10. og 90. prosentandelen i fordelingene.

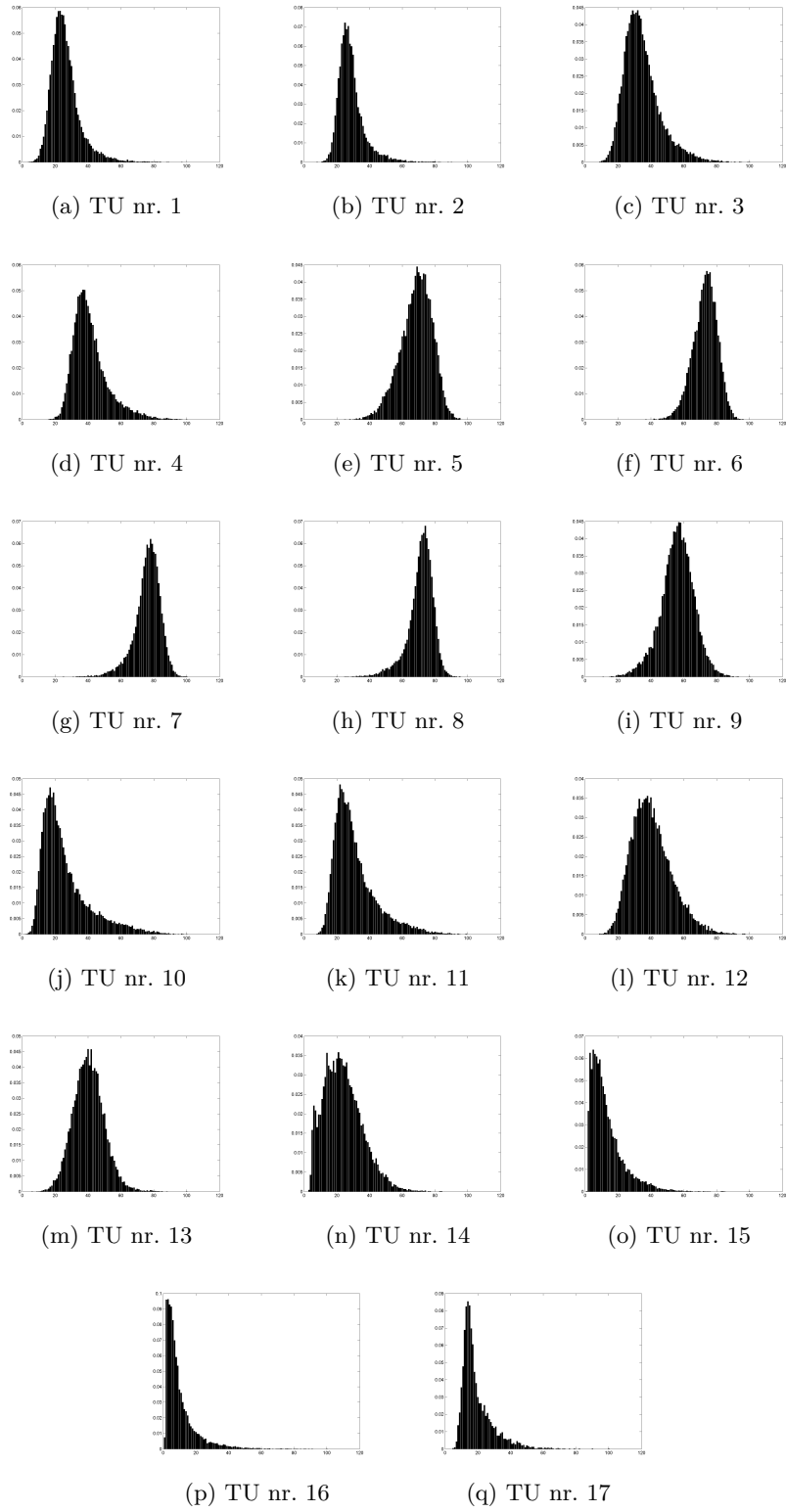
Nielsen og Danielsen [50] oppnådde gode resultater med adaptive egenskaper, hvor egenskapene blant annet ble basert på GLCM. Et forslag til videre arbeid er å beregne TU-bilder på tilsvarende måte som i denne oppgaven, for deretter å beregne GLCM fra disse TU-bildene. Ershad [19] har vært inne på liknende fremgangsmåte i klassifisering av steintekstur. Ved bruk av ulike klassifikatorer oppnådde Ershad en korrekt klassifiseringsrate mellom 90 og 94 %. Til tross for at resultatene er oppnådd på et annet felt, er resultatene lovende for bruk på andre områder. Sujatha et al. har også kombinert GLCM og LBP, men på en helt annerledes måte [60]. På et utvalg av teksturer fra Brodatz [7] oppnådde Sujatha et al. en gjennomsnittlig riktig klassifikasjon på 95 %. I [50] oppnådde Nielsen og Danielsen også gode resultater ved å basere de adaptive egenskapene på GLRLM og CGLRLM. En annen mulighet kan derfor være å beregne henholdsvis GLRLM og CGLRLM fra TU-bildene, i stedet for GLCM.

Kapittel 7

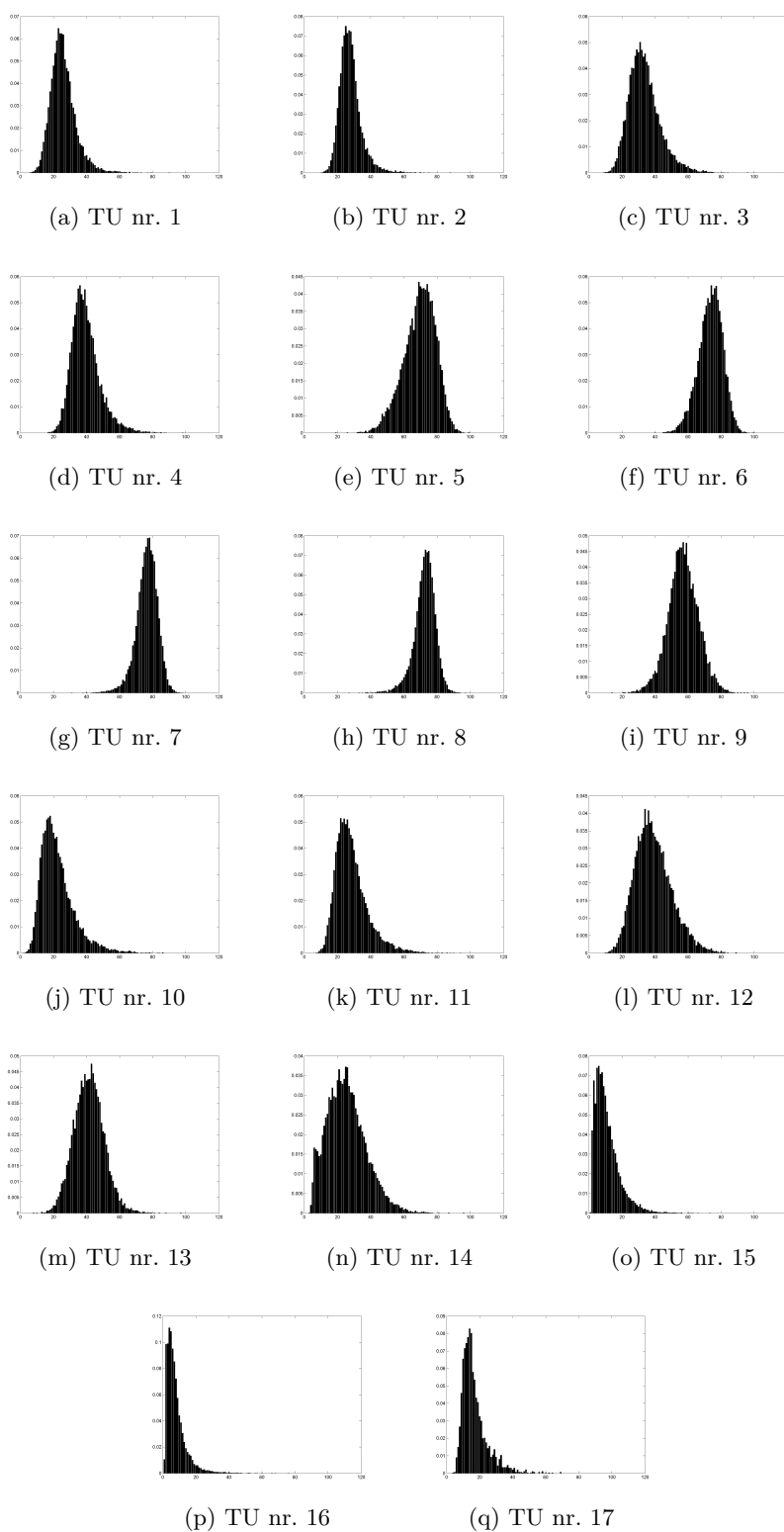
Vedlegg

7.1 Vedlegg A

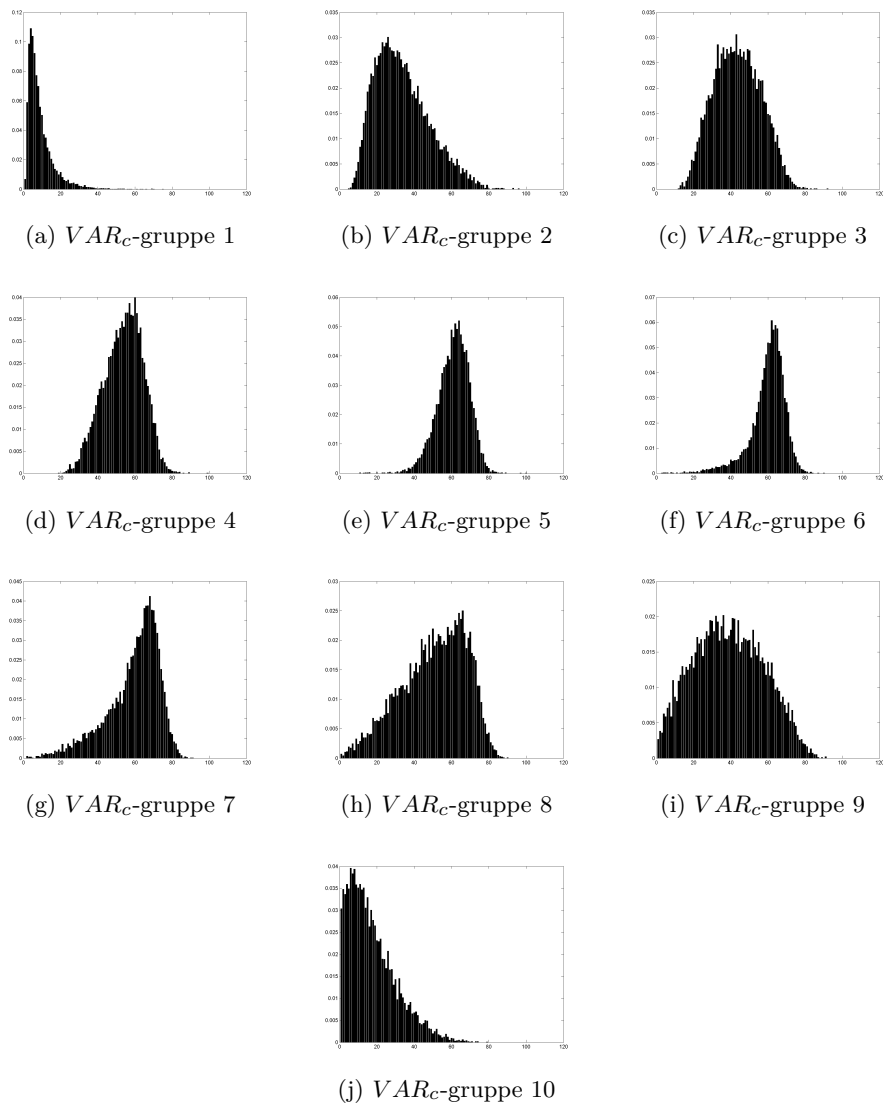
Dette vedlegget inneholder sannsynlighetsfordelingene til de 17 TU-gruppene og de 10 VAR_c -gruppene, for henholdsvis god og dårlig prognose. Histogrammene representerer pasientene i treningsettet til M98.



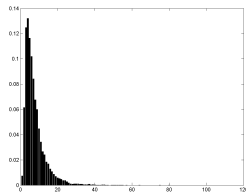
Figur 7.1: Sannsynlighetsfordelinger for pasientgruppe 0, beregnet fra de originale bildene med 8 naboer. Pasientene er hentet fra treningssettet til M98.



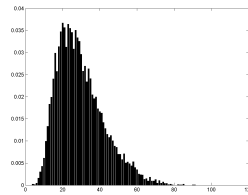
Figur 7.2: Sannsynlighetsfordelinger for gruppe 1, beregnet fra de originale bildene med 8 naboer. Pasientene er hentet fra treningssettet til M98.



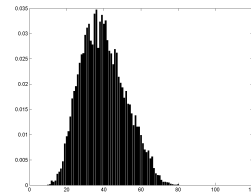
Figur 7.3: Sannsynlighetsfordelinger for alle de 10 VAR_c -gruppene for pasientene med god prognose, treningsett 1. Dataene er hentet fra M98.



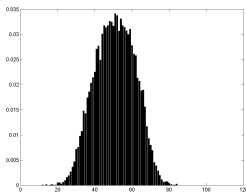
(a) VAR_c -gruppe 1



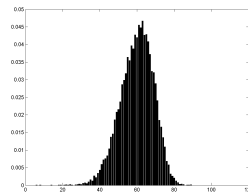
(b) VAR_c -gruppe 2



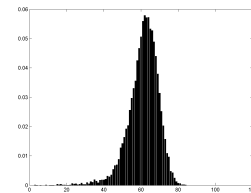
(c) VAR_c -gruppe 3



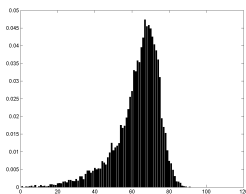
(d) VAR_c -gruppe 4



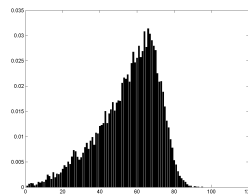
(e) VAR_c -gruppe 5



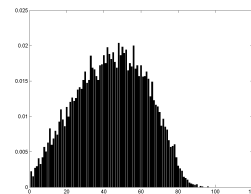
(f) VAR_c -gruppe 6



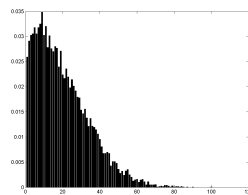
(g) VAR_c -gruppe 7



(h) VAR_c -gruppe 8



(i) VAR_c -gruppe 9



(j) VAR_c -gruppe 10

Figur 7.4: Sannsynlighetsfordelinger for alle de 10 VAR_c -gruppene for pasientene med dårlig prognose, treningsett 1. Dataene er hentet fra M98.

7.2 Vedlegg B

Dette vedlegget inneholder resultatene fra datasettet M98. Resultatene består av forvirringsmatriser, sensitivitet, spesifisitet og CCR_{eq}. For datasettet finner man resultatene fra bruk av operatorer med 8 og 16 piksels nabolag benyttet på de originale bildene, samt bruk av operatorer med 8 piksels nabolag benyttet på de nedskalerte bildene.

Tabell 7.1: Denne tabellen viser treningsresultatet for $LBP_{8,1}$ benyttet på de originale bildene. Dataene er hentet fra M98. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	5	5	10	Spesifisitet:	50 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	6	14	20	CCR _{eq} :	70 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	75 %
Gruppe 0:	5	5	10	Spesifisitet:	50 %
Gruppe 1:	0	10	10	Sensitivitet:	100 %
Totalt:	5	15	20	CCR _{eq} :	75 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	75 %
Gruppe 0:	5	5	10	Spesifisitet:	50 %
Gruppe 1:	0	10	10	Sensitivitet:	100 %
Totalt:	5	15	20	CCR _{eq} :	75 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	75 %
Gruppe 0:	5	5	10	Spesifisitet:	50 %
Gruppe 1:	0	10	10	Sensitivitet:	100 %
Totalt:	5	15	20	CCR _{eq} :	75 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	50 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	2	8	10	Sensitivitet:	80 %
Totalt:	4	16	20	CCR _{eq} :	50 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	3	7	10	Spesifisitet:	30 %
Gruppe 1:	2	8	10	Sensitivitet:	80 %
Totalt:	5	15	20	CCR _{eq} :	55 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	5	15	20	CCR _{eq} :	65 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	75 %
Gruppe 0:	5	5	10	Spesifisitet:	50 %
Gruppe 1:	0	10	10	Sensitivitet:	100 %
Totalt:	5	15	20	CCR _{eq} :	75 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	80 %
Gruppe 0:	6	4	10	Spesifisitet:	60 %
Gruppe 1:	0	10	10	Sensitivitet:	100 %
Totalt:	6	14	20	CCR _{eq} :	80 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	5	5	10	Spesifisitet:	50 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	6	14	20	CCR _{eq} :	70 %
Nr. 11	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	75 %
Gruppe 0:	5	5	10	Spesifisitet:	50 %
Gruppe 1:	0	10	10	Sensitivitet:	100 %
Totalt:	5	15	20	CCR _{eq} :	75 %
Nr. 12	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	5	15	20	CCR _{eq} :	65 %
Nr. 13	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	3	7	10	Spesifisitet:	30 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	4	16	20	CCR _{eq} :	60 %
Nr. 14	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	3	17	20	CCR _{eq} :	55 %
Nr. 15	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	5	15	20	CCR _{eq} :	65 %
Nr. 16	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	6	4	10	Spesifisitet:	60 %
Gruppe 1:	2	8	10	Sensitivitet:	80 %
Totalt:	8	12	20	CCR _{eq} :	70 %
Nr. 17	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	3	7	10	Spesifisitet:	30 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	4	16	20	CCR _{eq} :	60 %

Tabell 7.2: Denne tabellen viser testresultatet for $LBP_{8,1}$ benyttet på de originale bildene. Dataene er hentet fra M98. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	14 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	5	1	6	Sensitivitet:	17 %
Totalt:	6	8	14	CCR _{eq} :	15 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	14 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	5	1	6	Sensitivitet:	17 %
Totalt:	6	8	14	CCR _{eq} :	15 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	14 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	5	1	6	Sensitivitet:	17 %
Totalt:	6	8	14	CCR _{eq} :	15 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	21 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	5	1	6	Sensitivitet:	17 %
Totalt:	7	7	14	CCR _{eq} :	21 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	2	12	14	CCR _{eq} :	33 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	21 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	3	3	6	Sensitivitet:	50 %
Totalt:	3	11	14	CCR _{eq} :	25 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	1	5	6	Sensitivitet:	83 %
Totalt:	1	13	14	CCR _{eq} :	42 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	4	2	6	Sensitivitet:	33 %
Totalt:	6	8	14	CCR _{eq} :	29 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	21 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	5	1	6	Sensitivitet:	17 %
Totalt:	7	7	14	CCR _{eq} :	21 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	3	3	6	Sensitivitet:	50 %
Totalt:	4	10	14	CCR _{eq} :	31 %
Nr. 11	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	21 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	4	2	6	Sensitivitet:	33 %
Totalt:	5	9	14	CCR _{eq} :	23 %
Nr. 12	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	14 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	5	1	6	Sensitivitet:	17 %
Totalt:	6	8	14	CCR _{eq} :	15 %
Nr. 13	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	2	12	14	CCR _{eq} :	33 %
Nr. 14	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	1	5	6	Sensitivitet:	83 %
Totalt:	1	13	14	CCR _{eq} :	42 %
Nr. 15	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	21 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	3	3	6	Sensitivitet:	50 %
Totalt:	3	11	14	CCR _{eq} :	25 %
Nr. 16	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	3	3	6	Sensitivitet:	50 %
Totalt:	4	10	14	CCR _{eq} :	31 %
Nr. 17	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	50 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	1	5	6	Sensitivitet:	83 %
Totalt:	3	11	14	CCR _{eq} :	54 %

Tabell 7.3: Denne tabellen viser treningsresultatet for VAR_c benyttet på de originale bildene. Dataene er hentet fra M98. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	6	4	10	Spesifisitet:	60%
Gruppe 1:	5	5	10	Sensitivitet:	50%
Totalt:	11	9	20	CCR _{eq} :	55 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	4	6	10	Spesifisitet:	40%
Gruppe 1:	2	8	10	Sensitivitet:	80%
Totalt:	6	14	20	CCR _{eq} :	60 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	5	5	10	Spesifisitet:	50%
Gruppe 1:	3	7	10	Sensitivitet:	70%
Totalt:	8	12	20	CCR _{eq} :	60 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	6	4	10	Spesifisitet:	60%
Gruppe 1:	2	8	10	Sensitivitet:	80%
Totalt:	8	12	20	CCR _{eq} :	70 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	3	7	10	Spesifisitet:	30%
Gruppe 1:	1	9	10	Sensitivitet:	90%
Totalt:	4	16	20	CCR _{eq} :	60 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	6	4	10	Spesifisitet:	60%
Gruppe 1:	3	7	10	Sensitivitet:	70%
Totalt:	9	11	20	CCR _{eq} :	65 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	8	2	10	Spesifisitet:	80%
Gruppe 1:	6	4	10	Sensitivitet:	40%
Totalt:	14	6	20	CCR _{eq} :	60 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	8	2	10	Spesifisitet:	80%
Gruppe 1:	4	6	10	Sensitivitet:	60%
Totalt:	12	8	20	CCR _{eq} :	70 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	6	4	10	Spesifisitet:	60%
Gruppe 1:	4	6	10	Sensitivitet:	60%
Totalt:	10	10	20	CCR _{eq} :	60 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	4	6	10	Spesifisitet:	40%
Gruppe 1:	1	9	10	Sensitivitet:	90%
Totalt:	5	15	20	CCR _{eq} :	65 %

Tabell 7.4: Denne tabellen viser testresultatet for VAR_c benyttet på de originale bildene. Dataene er hentet fra M98. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	4	2	6	Sensitivitet:	33 %
Totalt:	6	8	14	CCR _{eq} :	29 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	3	3	6	Sensitivitet:	50 %
Totalt:	4	10	14	CCR _{eq} :	31 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	3	11	14	CCR _{eq} :	40 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	3	5	8	Spesifisitet:	38 %
Gruppe 1:	3	3	6	Sensitivitet:	50 %
Totalt:	6	8	14	CCR _{eq} :	44 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	3	11	14	CCR _{eq} :	40 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	3	3	6	Sensitivitet:	50 %
Totalt:	4	10	14	CCR _{eq} :	31 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	3	5	8	Spesifisitet:	38 %
Gruppe 1:	5	1	6	Sensitivitet:	17 %
Totalt:	8	6	14	CCR _{eq} :	27 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	4	2	6	Sensitivitet:	33 %
Totalt:	6	8	14	CCR _{eq} :	29 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	3	3	6	Sensitivitet:	50 %
Totalt:	4	10	14	CCR _{eq} :	31 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	3	11	14	CCR _{eq} :	40 %

Tabell 7.5: Denne tabellen viser treningsresultatet for $LBP_{16,2}$ benyttet på de originale bildene. Dataene er hentet fra M98. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCRReq.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	3	17	20	CCRReq:	55 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	3	7	10	Spesifisitet:	30 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	4	16	20	CCRReq:	60 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	2	8	10	Sensitivitet:	80 %
Totalt:	6	14	20	CCRReq:	60 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	5	15	20	CCRReq:	65 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	3	17	20	CCRReq:	55 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	3	17	20	CCRReq:	55 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	5	15	20	CCRReq:	65 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	9	1	10	Spesifisitet:	90 %
Gruppe 1:	7	3	10	Sensitivitet:	30 %
Totalt:	16	4	20	CCRReq:	60 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	5	5	10	Spesifisitet:	50 %
Gruppe 1:	4	6	10	Sensitivitet:	60 %
Totalt:	9	11	20	CCRReq:	55 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	5	15	20	CCRReq:	65 %
Nr. 11	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	5	15	20	CCRReq:	65 %
Nr. 12	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	75 %
Gruppe 0:	6	4	10	Spesifisitet:	60 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	7	13	20	CCRReq:	75 %
Nr. 13	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	6	4	10	Spesifisitet:	60 %
Gruppe 1:	2	8	10	Sensitivitet:	80 %
Totalt:	8	12	20	CCRReq:	70 %
Nr. 14	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	0	10	10	Sensitivitet:	100 %
Totalt:	4	16	20	CCRReq:	70 %
Nr. 15	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	3	17	20	CCRReq:	55 %
Nr. 16	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	3	17	20	CCRReq:	55 %
Nr. 17	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	45 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	3	7	10	Sensitivitet:	70 %
Totalt:	5	15	20	CCRReq:	45 %
Nr. 18	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	5	5	10	Spesifisitet:	50 %
Gruppe 1:	2	8	10	Sensitivitet:	80 %
Totalt:	7	13	20	CCRReq:	65 %

Tabell 7.6: Denne tabellen viser testresultatet for $LBP_{16,2}$ benyttet på de originale bildene. Dataene er hentet fra M98. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	57 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	2	12	14	CCR _{eq} :	63 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	1	5	6	Sensitivitet:	83 %
Totalt:	1	13	14	CCR _{eq} :	42 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	3	3	6	Sensitivitet:	50 %
Totalt:	4	10	14	CCR _{eq} :	31 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	1	5	6	Sensitivitet:	83 %
Totalt:	1	13	14	CCR _{eq} :	42 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	0	14	14	CCR _{eq} :	50 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	0	14	14	CCR _{eq} :	50 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	1	5	6	Sensitivitet:	83 %
Totalt:	1	13	14	CCR _{eq} :	42 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	3	5	8	Spesifisitet:	38 %
Gruppe 1:	3	3	6	Sensitivitet:	50 %
Totalt:	6	8	14	CCR _{eq} :	44 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	2	12	14	CCR _{eq} :	33 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	1	5	6	Sensitivitet:	83 %
Totalt:	1	13	14	CCR _{eq} :	42 %
Nr. 11	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	1	5	6	Sensitivitet:	83 %
Totalt:	1	13	14	CCR _{eq} :	42 %
Nr. 12	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	2	12	14	CCR _{eq} :	33 %
Nr. 13	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	3	3	6	Sensitivitet:	50 %
Totalt:	5	9	14	CCR _{eq} :	38 %
Nr. 14	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	50 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	1	5	6	Sensitivitet:	83 %
Totalt:	3	11	14	CCR _{eq} :	54 %
Nr. 15	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	0	14	14	CCR _{eq} :	50 %
Nr. 16	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	1	5	6	Sensitivitet:	83 %
Totalt:	1	13	14	CCR _{eq} :	42 %
Nr. 17	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	4	2	6	Sensitivitet:	33 %
Totalt:	6	8	14	CCR _{eq} :	29 %
Nr. 18	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	21 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	4	2	6	Sensitivitet:	33 %
Totalt:	5	9	14	CCR _{eq} :	23 %

Tabell 7.7: Denne tabellen viser treningsresultatet for $VAR_{c16,2}$ benyttet på de originale bildene. Dataene er hentet fra M98. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	3	7	10	Sensitivitet:	70 %
Totalt:	7	13	20	CCR _{eq} :	55 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	6	4	10	Spesifisitet:	60 %
Gruppe 1:	5	5	10	Sensitivitet:	50 %
Totalt:	11	9	20	CCR _{eq} :	55 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	8	2	10	Spesifisitet:	80 %
Gruppe 1:	7	3	10	Sensitivitet:	30 %
Totalt:	15	5	20	CCR _{eq} :	55 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	8	2	10	Spesifisitet:	80 %
Gruppe 1:	7	3	10	Sensitivitet:	30 %
Totalt:	15	5	20	CCR _{eq} :	55 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	9	1	10	Spesifisitet:	90 %
Gruppe 1:	5	5	10	Sensitivitet:	50 %
Totalt:	14	6	20	CCR _{eq} :	70 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	7	3	10	Spesifisitet:	70 %
Gruppe 1:	6	4	10	Sensitivitet:	40 %
Totalt:	13	7	20	CCR _{eq} :	55 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	50 %
Gruppe 0:	8	2	10	Spesifisitet:	80 %
Gruppe 1:	8	2	10	Sensitivitet:	20 %
Totalt:	16	4	20	CCR _{eq} :	50 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	7	3	10	Spesifisitet:	70 %
Gruppe 1:	6	4	10	Sensitivitet:	40 %
Totalt:	13	7	20	CCR _{eq} :	55 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	8	2	10	Spesifisitet:	80 %
Gruppe 1:	7	3	10	Sensitivitet:	30 %
Totalt:	15	5	20	CCR _{eq} :	55 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	8	2	10	Spesifisitet:	80 %
Gruppe 1:	5	5	10	Sensitivitet:	50 %
Totalt:	13	7	20	CCR _{eq} :	65 %

Tabell 7.8: Denne tabellen viser testresultatet for $VAR_{c16,2}$ benyttet på de originale bildene. Dataene er hentet fra M98. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCRq.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	3	11	14	CCRq:	40 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	4	2	6	Sensitivitet:	33 %
Totalt:	6	8	14	CCRq:	29 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	21 %
Gruppe 0:	3	5	8	Spesifisitet:	38 %
Gruppe 1:	6	0	6	Sensitivitet:	0 %
Totalt:	9	5	14	CCRq:	19 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	4	4	8	Spesifisitet:	50 %
Gruppe 1:	5	1	6	Sensitivitet:	17 %
Totalt:	9	5	14	CCRq:	33 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	4	4	8	Spesifisitet:	50 %
Gruppe 1:	5	1	6	Sensitivitet:	17 %
Totalt:	9	5	14	CCRq:	33 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	86 %
Gruppe 0:	8	0	8	Spesifisitet:	100 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	10	4	14	CCRq:	83 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	50 %
Gruppe 0:	7	1	8	Spesifisitet:	88 %
Gruppe 1:	6	0	6	Sensitivitet:	0 %
Totalt:	13	1	14	CCRq:	44 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	21 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	5	1	6	Sensitivitet:	17 %
Totalt:	7	7	14	CCRq:	21 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	4	4	8	Spesifisitet:	50 %
Gruppe 1:	6	0	6	Sensitivitet:	0 %
Totalt:	10	4	14	CCRq:	25 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	5	3	8	Spesifisitet:	63 %
Gruppe 1:	5	1	6	Sensitivitet:	17 %
Totalt:	10	4	14	CCRq:	40 %

Tabell 7.9: Denne tabellen viser treningsresultatet for $LBP_{8,1}$ benyttet på de nedskalerte bildene. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	5	15	20	CCR _{eq} :	65 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	3	17	20	CCR _{eq} :	55 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	3	17	20	CCR _{eq} :	55 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	3	17	20	CCR _{eq} :	55 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	3	7	10	Spesifisitet:	30 %
Gruppe 1:	2	8	10	Sensitivitet:	80 %
Totalt:	5	15	20	CCR _{eq} :	55 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	5	5	10	Spesifisitet:	50 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	6	14	20	CCR _{eq} :	70 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	3	7	10	Spesifisitet:	30 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	4	16	20	CCR _{eq} :	60 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	3	17	20	CCR _{eq} :	55 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	5	15	20	CCR _{eq} :	65 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	3	17	20	CCR _{eq} :	55 %
Nr. 11	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	3	17	20	CCR _{eq} :	55 %
Nr. 12	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	3	17	20	CCR _{eq} :	55 %
Nr. 13	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	5	15	20	CCR _{eq} :	65 %
Nr. 14	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	4	6	10	Spesifisitet:	40 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	5	15	20	CCR _{eq} :	65 %
Nr. 15	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	5	5	10	Spesifisitet:	50 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	6	14	20	CCR _{eq} :	70 %
Nr. 16	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	0	10	10	Sensitivitet:	100 %
Totalt:	2	18	20	CCR _{eq} :	60 %
Nr. 17	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	0 %
Gruppe 0:	0	10	10	Spesifisitet:	0 %
Gruppe 1:	10	0	10	Sensitivitet:	0 %
Totalt:	10	10	20	CCR _{eq} :	0 %

Tabell 7.10: Denne tabellen viser testresultatet for $LBP_{8,1}$ benyttet på de nedskalerte bildene. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	57 %
Gruppe 0:	2	6	8	Spesifisitet:	25%
Gruppe 1:	0	6	6	Sensitivitet:	100%
Totalt:	2	12	14	CCR _{eq} :	63 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	0	14	14	CCR _{eq} :	50 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	57 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	2	12	14	CCR _{eq} :	63 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	57 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	2	12	14	CCR _{eq} :	63 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	57 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	2	12	14	CCR _{eq} :	63 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	64 %
Gruppe 0:	3	5	8	Spesifisitet:	38 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	3	11	14	CCR _{eq} :	69 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	0	14	14	CCR _{eq} :	50 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	0	14	14	CCR _{eq} :	50 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	57 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	2	12	14	CCR _{eq} :	63 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	0	14	14	CCR _{eq} :	50 %
Nr. 11	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	0	14	14	CCR _{eq} :	50 %
Nr. 12	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	57 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	0	6	6	Sensitivitet:	100 %
Totalt:	2	12	14	CCR _{eq} :	63 %
Nr. 13	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	50 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	1	5	6	Sensitivitet:	83 %
Totalt:	3	11	14	CCR _{eq} :	54 %
Nr. 14	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	57 %
Gruppe 0:	3	5	8	Spesifisitet:	38 %
Gruppe 1:	1	5	6	Sensitivitet:	83 %
Totalt:	4	10	14	CCR _{eq} :	60 %
Nr. 15	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	3	11	14	CCR _{eq} :	40 %
Nr. 16	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	1	5	6	Sensitivitet:	83 %
Totalt:	1	13	14	CCR _{eq} :	42 %
Nr. 17	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	0 %
Gruppe 0:	0	8	8	Spesifisitet:	0 %
Gruppe 1:	6	0	6	Sensitivitet:	0 %
Totalt:	6	8	14	CCR _{eq} :	0 %

Tabell 7.11: Denne tabellen viser treningsresultatet for VAR_c benyttet på de nedskalerte bildene. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	50 %
Gruppe 0:	2	8	10	Spesifisitet:	20 %
Gruppe 1:	2	8	10	Sensitivitet:	80 %
Totalt:	4	16	20	CCR _{eq} :	50 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	45 %
Gruppe 0:	5	5	10	Spesifisitet:	50 %
Gruppe 1:	6	4	10	Sensitivitet:	40 %
Totalt:	11	9	20	CCR _{eq} :	45 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	8	2	10	Spesifisitet:	80 %
Gruppe 1:	4	6	10	Sensitivitet:	60 %
Totalt:	12	8	20	CCR _{eq} :	70 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	8	2	10	Spesifisitet:	80 %
Gruppe 1:	5	5	10	Sensitivitet:	50 %
Totalt:	13	7	20	CCR _{eq} :	65 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	6	4	10	Spesifisitet:	60 %
Gruppe 1:	5	5	10	Sensitivitet:	50 %
Totalt:	11	9	20	CCR _{eq} :	55 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	3	7	10	Spesifisitet:	30 %
Gruppe 1:	2	8	10	Sensitivitet:	80 %
Totalt:	5	15	20	CCR _{eq} :	55 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	3	7	10	Spesifisitet:	30 %
Gruppe 1:	1	9	10	Sensitivitet:	90 %
Totalt:	4	16	20	CCR _{eq} :	60 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	50 %
Gruppe 0:	6	4	10	Spesifisitet:	60 %
Gruppe 1:	6	4	10	Sensitivitet:	40 %
Totalt:	12	8	20	CCR _{eq} :	50 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	8	2	10	Spesifisitet:	80 %
Gruppe 1:	6	4	10	Sensitivitet:	40 %
Totalt:	14	6	20	CCR _{eq} :	60 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	65 %
Gruppe 0:	7	3	10	Spesifisitet:	70 %
Gruppe 1:	4	6	10	Sensitivitet:	60 %
Totalt:	11	9	20	CCR _{eq} :	65 %

Tabell 7.12: Denne tabellen viser testresultatet for VAR_c benyttet på de nedskalerte bildene. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet $CCReq$.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	1	7	8	Spesifisitet:	13 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	3	11	14	CCReq:	40 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	3	3	6	Sensitivitet:	50 %
Totalt:	5	9	14	CCReq:	38 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	4	4	8	Spesifisitet:	50 %
Gruppe 1:	5	1	6	Sensitivitet:	17 %
Totalt:	9	5	14	CCReq:	33 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	64 %
Gruppe 0:	7	1	8	Spesifisitet:	88 %
Gruppe 1:	4	2	6	Sensitivitet:	33 %
Totalt:	11	3	14	CCReq:	60 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	57 %
Gruppe 0:	4	4	8	Spesifisitet:	50 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	6	8	14	CCReq:	58 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	50 %
Gruppe 0:	3	5	8	Spesifisitet:	38 %
Gruppe 1:	2	4	6	Sensitivitet:	67 %
Totalt:	5	9	14	CCReq:	52 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	3	3	6	Sensitivitet:	50 %
Totalt:	5	9	14	CCReq:	38 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	2	6	8	Spesifisitet:	25 %
Gruppe 1:	4	2	6	Sensitivitet:	33 %
Totalt:	6	8	14	CCReq:	29 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	29 %
Gruppe 0:	4	4	8	Spesifisitet:	50 %
Gruppe 1:	6	0	6	Sensitivitet:	0 %
Totalt:	10	4	14	CCReq:	25 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	57 %
Gruppe 0:	6	2	8	Spesifisitet:	75 %
Gruppe 1:	4	2	6	Sensitivitet:	33 %
Totalt:	10	4	14	CCReq:	54 %

7.3 Vedlegg C

Dette vedlegget inneholder resultatene fra datasettet L23. Resultatene består av forvirringsmatriser, sensitivitet, spesifisitet og CCR_{eq}. For datasettet finner man resultatene fra bruk av operatorer med 8 og 16 piksels nabolag benyttet på de originale bildene, samt bruk av operatorer med 8 piksels nabolag benyttet på de nedskalerte bildene.

Tabell 7.13: Denne tabellen viser treningsresultatet for $LBP_{8,1}$ benyttet på de originale bildene i L23. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	58 %
Gruppe 0:	27	22	49	Spesifisitet:	55 %
Gruppe 1:	6	12	18	Sensitivitet:	67 %
Totalt:	33	34	67	CCR _{eq} :	61 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	17	32	49	Spesifisitet:	35 %
Gruppe 1:	6	12	18	Sensitivitet:	67 %
Totalt:	23	44	67	CCR _{eq} :	51 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	58 %
Gruppe 0:	28	21	49	Spesifisitet:	57 %
Gruppe 1:	7	11	18	Sensitivitet:	61 %
Totalt:	35	32	67	CCR _{eq} :	59 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	49 %
Gruppe 0:	19	30	49	Spesifisitet:	39 %
Gruppe 1:	4	14	18	Sensitivitet:	78 %
Totalt:	23	44	67	CCR _{eq} :	58 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	37 %
Gruppe 0:	12	37	49	Spesifisitet:	24 %
Gruppe 1:	5	13	18	Sensitivitet:	72 %
Totalt:	17	50	67	CCR _{eq} :	48 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	67 %
Gruppe 0:	36	13	49	Spesifisitet:	73 %
Gruppe 1:	9	9	18	Sensitivitet:	50 %
Totalt:	45	22	67	CCR _{eq} :	62 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	45 %
Gruppe 0:	17	32	49	Spesifisitet:	35 %
Gruppe 1:	5	13	18	Sensitivitet:	72 %
Totalt:	22	45	67	CCR _{eq} :	53 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	58 %
Gruppe 0:	26	23	49	Spesifisitet:	53 %
Gruppe 1:	5	13	18	Sensitivitet:	72 %
Totalt:	31	36	67	CCR _{eq} :	63 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	42 %
Gruppe 0:	14	35	49	Spesifisitet:	29 %
Gruppe 1:	4	14	18	Sensitivitet:	78 %
Totalt:	18	49	67	CCR _{eq} :	53 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	58 %
Gruppe 0:	24	25	49	Spesifisitet:	49 %
Gruppe 1:	3	15	18	Sensitivitet:	83 %
Totalt:	27	40	67	CCR _{eq} :	66 %
Nr. 11	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	61 %
Gruppe 0:	28	21	49	Spesifisitet:	57 %
Gruppe 1:	5	13	18	Sensitivitet:	72 %
Totalt:	33	34	67	CCR _{eq} :	65 %
Nr. 12	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	42 %
Gruppe 0:	14	35	49	Spesifisitet:	29 %
Gruppe 1:	4	14	18	Sensitivitet:	78 %
Totalt:	18	49	67	CCR _{eq} :	53 %
Nr. 13	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	49 %
Gruppe 0:	20	29	49	Spesifisitet:	41 %
Gruppe 1:	5	13	18	Sensitivitet:	72 %
Totalt:	25	42	67	CCR _{eq} :	57 %
Nr. 14	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	52 %
Gruppe 0:	22	27	49	Spesifisitet:	45 %
Gruppe 1:	5	13	18	Sensitivitet:	72 %
Totalt:	27	40	67	CCR _{eq} :	59 %
Nr. 15	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	51 %
Gruppe 0:	21	28	49	Spesifisitet:	43 %
Gruppe 1:	5	13	18	Sensitivitet:	72 %
Totalt:	26	41	67	CCR _{eq} :	58 %
Nr. 16	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	29	20	49	Spesifisitet:	59 %
Gruppe 1:	7	11	18	Sensitivitet:	61 %
Totalt:	36	31	67	CCR _{eq} :	60 %
Nr. 17	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	66 %
Gruppe 0:	27	22	49	Spesifisitet:	55 %
Gruppe 1:	1	17	18	Sensitivitet:	94 %
Totalt:	28	39	67	CCR _{eq} :	75 %

Tabell 7.14: Denne tabellen viser treningsresultatet for $LBP_{8,1}$ benyttet på de originale bildene i L23. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	48 %
Gruppe 0:	22	26	48	Spesifisitet:	46 %
Gruppe 1:	9	10	19	Sensitivitet:	53 %
Totalt:	31	36	67	CCR _{eq} :	49 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	42 %
Gruppe 0:	17	31	48	Spesifisitet:	35 %
Gruppe 1:	8	11	19	Sensitivitet:	58 %
Totalt:	25	42	67	CCR _{eq} :	47 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	46 %
Gruppe 0:	22	26	48	Spesifisitet:	46 %
Gruppe 1:	10	9	19	Sensitivitet:	47 %
Totalt:	32	35	67	CCR _{eq} :	47 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	42 %
Gruppe 0:	18	30	48	Spesifisitet:	38 %
Gruppe 1:	9	10	19	Sensitivitet:	53 %
Totalt:	27	40	67	CCR _{eq} :	45 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	31 %
Gruppe 0:	9	39	48	Spesifisitet:	19 %
Gruppe 1:	7	12	19	Sensitivitet:	63 %
Totalt:	16	51	67	CCR _{eq} :	41 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	67 %
Gruppe 0:	36	12	48	Spesifisitet:	75 %
Gruppe 1:	10	9	19	Sensitivitet:	47 %
Totalt:	46	21	67	CCR _{eq} :	61 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	45 %
Gruppe 0:	17	31	48	Spesifisitet:	35 %
Gruppe 1:	6	13	19	Sensitivitet:	68 %
Totalt:	23	44	67	CCR _{eq} :	52 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	45 %
Gruppe 0:	21	27	48	Spesifisitet:	44 %
Gruppe 1:	10	9	19	Sensitivitet:	47 %
Totalt:	31	36	67	CCR _{eq} :	46 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	45 %
Gruppe 0:	18	30	48	Spesifisitet:	38 %
Gruppe 1:	7	12	19	Sensitivitet:	63 %
Totalt:	25	42	67	CCR _{eq} :	50 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	49 %
Gruppe 0:	23	25	48	Spesifisitet:	48 %
Gruppe 1:	9	10	19	Sensitivitet:	53 %
Totalt:	32	35	67	CCR _{eq} :	50 %
Nr. 11	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	46 %
Gruppe 0:	21	27	48	Spesifisitet:	44 %
Gruppe 1:	9	10	19	Sensitivitet:	53 %
Totalt:	30	37	67	CCR _{eq} :	48 %
Nr. 12	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	40 %
Gruppe 0:	17	31	48	Spesifisitet:	35 %
Gruppe 1:	9	10	19	Sensitivitet:	53 %
Totalt:	26	41	67	CCR _{eq} :	44 %
Nr. 13	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	49 %
Gruppe 0:	20	28	48	Spesifisitet:	42 %
Gruppe 1:	6	13	19	Sensitivitet:	68 %
Totalt:	26	41	67	CCR _{eq} :	55 %
Nr. 14	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	58 %
Gruppe 0:	25	23	48	Spesifisitet:	52 %
Gruppe 1:	5	14	19	Sensitivitet:	74 %
Totalt:	30	37	67	CCR _{eq} :	63 %
Nr. 15	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	18	30	48	Spesifisitet:	38 %
Gruppe 1:	8	11	19	Sensitivitet:	58 %
Totalt:	26	41	67	CCR _{eq} :	48 %
Nr. 16	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	28	20	48	Spesifisitet:	58 %
Gruppe 1:	10	9	19	Sensitivitet:	47 %
Totalt:	38	29	67	CCR _{eq} :	53 %
Nr. 17	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	45 %
Gruppe 0:	15	33	48	Spesifisitet:	31 %
Gruppe 1:	4	15	19	Sensitivitet:	79 %
Totalt:	19	48	67	CCR _{eq} :	55 %

Tabell 7.15: Denne tabellen viser treningsresultatet for VAR_c benyttet på de originale bildene i L23. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	13	36	49	Spesifisitet:	27 %
Gruppe 1:	2	16	18	Sensitivitet:	89 %
Totalt:	15	52	67	CCR _{eq} :	58 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	58 %
Gruppe 0:	31	18	49	Spesifisitet:	63 %
Gruppe 1:	10	8	18	Sensitivitet:	44 %
Totalt:	41	26	67	CCR _{eq} :	54 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	54 %
Gruppe 0:	25	24	49	Spesifisitet:	51 %
Gruppe 1:	7	11	18	Sensitivitet:	61 %
Totalt:	32	35	67	CCR _{eq} :	56 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	51 %
Gruppe 0:	18	31	49	Spesifisitet:	37 %
Gruppe 1:	2	16	18	Sensitivitet:	89 %
Totalt:	20	47	67	CCR _{eq} :	63 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	45 %
Gruppe 0:	14	35	49	Spesifisitet:	29 %
Gruppe 1:	2	16	18	Sensitivitet:	89 %
Totalt:	16	51	67	CCR _{eq} :	59 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	16	33	49	Spesifisitet:	33 %
Gruppe 1:	5	13	18	Sensitivitet:	72 %
Totalt:	21	46	67	CCR _{eq} :	52 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	33	16	49	Spesifisitet:	67 %
Gruppe 1:	11	7	18	Sensitivitet:	39 %
Totalt:	44	23	67	CCR _{eq} :	53 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	29	20	49	Spesifisitet:	59 %
Gruppe 1:	10	8	18	Sensitivitet:	44 %
Totalt:	39	28	67	CCR _{eq} :	52 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	61 %
Gruppe 0:	33	16	49	Spesifisitet:	67 %
Gruppe 1:	10	8	18	Sensitivitet:	44 %
Totalt:	43	24	67	CCR _{eq} :	56 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	52 %
Gruppe 0:	26	23	49	Spesifisitet:	53 %
Gruppe 1:	9	9	18	Sensitivitet:	50 %
Totalt:	35	32	67	CCR _{eq} :	52 %

Tabell 7.16: Denne tabellen viser testresultatet for VAR_c benyttet på de originale bildene i L23. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	30 %
Gruppe 0:	8	40	48	Spesifisitet:	17 %
Gruppe 1:	7	12	19	Sensitivitet:	63 %
Totalt:	15	52	67	CCR _{eq} :	40 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	63 %
Gruppe 0:	33	15	48	Spesifisitet:	69 %
Gruppe 1:	10	9	19	Sensitivitet:	47 %
Totalt:	43	24	67	CCR _{eq} :	58 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	26	22	48	Spesifisitet:	54 %
Gruppe 1:	8	11	19	Sensitivitet:	58 %
Totalt:	34	33	67	CCR _{eq} :	56 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	13	35	48	Spesifisitet:	27 %
Gruppe 1:	3	16	19	Sensitivitet:	84 %
Totalt:	16	51	67	CCR _{eq} :	56 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	39 %
Gruppe 0:	14	34	48	Spesifisitet:	29 %
Gruppe 1:	7	12	19	Sensitivitet:	63 %
Totalt:	21	46	67	CCR _{eq} :	46 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	45 %
Gruppe 0:	17	31	48	Spesifisitet:	35 %
Gruppe 1:	6	13	19	Sensitivitet:	68 %
Totalt:	23	44	67	CCR _{eq} :	52 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	64 %
Gruppe 0:	36	12	48	Spesifisitet:	75 %
Gruppe 1:	12	7	19	Sensitivitet:	37 %
Totalt:	48	19	67	CCR _{eq} :	56 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	58 %
Gruppe 0:	29	19	48	Spesifisitet:	60 %
Gruppe 1:	9	10	19	Sensitivitet:	53 %
Totalt:	38	29	67	CCR _{eq} :	57 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	63 %
Gruppe 0:	31	17	48	Spesifisitet:	65 %
Gruppe 1:	8	11	19	Sensitivitet:	58 %
Totalt:	39	28	67	CCR _{eq} :	61 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	52 %
Gruppe 0:	25	23	48	Spesifisitet:	52 %
Gruppe 1:	9	10	19	Sensitivitet:	53 %
Totalt:	34	33	67	CCR _{eq} :	52 %

Tabell 7.17: Resultater for hver TU-gruppe, etter bruk av $LBP_{16,2}$ på treningsettet i L23

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	45	4	49	Spesifisitet:	92 %
Gruppe 1:	17	1	18	Sensitivitet:	6 %
Totalt:	62	5	67	CCReq:	49 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	72 %
Gruppe 0:	43	6	49	Spesifisitet:	88 %
Gruppe 1:	13	5	18	Sensitivitet:	28 %
Totalt:	56	11	67	CCReq:	58 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	38	11	49	Spesifisitet:	76 %
Gruppe 1:	9	9	18	Sensitivitet:	50 %
Totalt:	47	20	67	CCReq:	64 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	45	4	49	Spesifisitet:	92 %
Gruppe 1:	17	1	18	Sensitivitet:	6 %
Totalt:	62	5	67	CCReq:	49 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	45	4	49	Spesifisitet:	92 %
Gruppe 1:	16	2	18	Sensitivitet:	11 %
Totalt:	61	6	67	CCReq:	51 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	73 %
Gruppe 0:	42	7	49	Spesifisitet:	86 %
Gruppe 1:	11	7	18	Sensitivitet:	39 %
Totalt:	53	14	67	CCReq:	62 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	41 %
Gruppe 0:	14	35	49	Spesifisitet:	29 %
Gruppe 1:	4	14	18	Sensitivitet:	78 %
Totalt:	18	49	67	CCReq:	53 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	52 %
Gruppe 0:	22	27	49	Spesifisitet:	45 %
Gruppe 1:	5	13	18	Sensitivitet:	72 %
Totalt:	27	40	67	CCReq:	59 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	75 %
Gruppe 0:	45	4	49	Spesifisitet:	92 %
Gruppe 1:	13	5	18	Sensitivitet:	28 %
Totalt:	58	9	67	CCReq:	60 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	43	6	49	Spesifisitet:	88 %
Gruppe 1:	15	3	18	Sensitivitet:	17 %
Totalt:	58	9	67	CCReq:	52 %
Nr. 11	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	67 %
Gruppe 0:	43	6	49	Spesifisitet:	88 %
Gruppe 1:	16	2	18	Sensitivitet:	11 %
Totalt:	59	8	67	CCReq:	49 %
Nr. 12	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	67 %
Gruppe 0:	34	15	49	Spesifisitet:	69 %
Gruppe 1:	7	11	18	Sensitivitet:	61 %
Totalt:	41	26	67	CCReq:	65 %
Nr. 13	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	64 %
Gruppe 0:	31	18	49	Spesifisitet:	63 %
Gruppe 1:	6	12	18	Sensitivitet:	67 %
Totalt:	37	30	67	CCReq:	65 %
Nr. 14	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	72 %
Gruppe 0:	44	5	49	Spesifisitet:	90 %
Gruppe 1:	14	4	18	Sensitivitet:	22 %
Totalt:	58	9	67	CCReq:	56 %
Nr. 15	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	45	4	49	Spesifisitet:	92 %
Gruppe 1:	17	1	18	Sensitivitet:	6 %
Totalt:	62	5	67	CCReq:	49 %
Nr. 16	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	72 %
Gruppe 0:	43	6	49	Spesifisitet:	88 %
Gruppe 1:	13	5	18	Sensitivitet:	28 %
Totalt:	56	11	67	CCReq:	58 %
Nr. 17	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	38 %
Gruppe 0:	10	39	49	Spesifisitet:	20 %
Gruppe 1:	2	16	18	Sensitivitet:	89 %
Totalt:	12	55	67	CCReq:	55 %
Nr. 18	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	51 %
Gruppe 0:	22	27	49	Spesifisitet:	45 %
Gruppe 1:	6	12	18	Sensitivitet:	67 %
Totalt:	28	39	67	CCReq:	56 %

Tabell 7.18: Resultater for hver TU-gruppe, etter bruk av $LBP_{16,2}$ på testsettet i L23

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	72 %
Gruppe 0:	45	3	48	Spesifisitet:	94 %
Gruppe 1:	16	3	19	Sensitivitet:	16 %
Totalt:	61	6	67	CCReq:	55 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	66 %
Gruppe 0:	40	8	48	Spesifisitet:	83 %
Gruppe 1:	15	4	19	Sensitivitet:	21 %
Totalt:	55	12	67	CCReq:	52 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	37	11	48	Spesifisitet:	77 %
Gruppe 1:	10	9	19	Sensitivitet:	47 %
Totalt:	47	20	67	CCReq:	62 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	45	3	48	Spesifisitet:	94 %
Gruppe 1:	18	1	19	Sensitivitet:	5 %
Totalt:	63	4	67	CCReq:	50 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	66 %
Gruppe 0:	43	5	48	Spesifisitet:	90 %
Gruppe 1:	18	1	19	Sensitivitet:	5 %
Totalt:	61	6	67	CCReq:	47 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	67 %
Gruppe 0:	40	8	48	Spesifisitet:	83 %
Gruppe 1:	14	5	19	Sensitivitet:	26 %
Totalt:	54	13	67	CCReq:	55 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	43 %
Gruppe 0:	13	35	48	Spesifisitet:	27 %
Gruppe 1:	3	16	19	Sensitivitet:	84 %
Totalt:	16	51	67	CCReq:	56 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	22	26	48	Spesifisitet:	46 %
Gruppe 1:	4	15	19	Sensitivitet:	79 %
Totalt:	26	41	67	CCReq:	62 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	43	5	48	Spesifisitet:	90 %
Gruppe 1:	16	3	19	Sensitivitet:	16 %
Totalt:	59	8	67	CCReq:	53 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	64 %
Gruppe 0:	40	8	48	Spesifisitet:	83 %
Gruppe 1:	16	3	19	Sensitivitet:	16 %
Totalt:	56	11	67	CCReq:	50 %
Nr. 11	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	42	6	48	Spesifisitet:	88 %
Gruppe 1:	15	4	19	Sensitivitet:	21 %
Totalt:	57	10	67	CCReq:	54 %
Nr. 12	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	61 %
Gruppe 0:	34	14	48	Spesifisitet:	71 %
Gruppe 1:	12	7	19	Sensitivitet:	37 %
Totalt:	46	21	67	CCReq:	54 %
Nr. 13	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	63 %
Gruppe 0:	31	17	48	Spesifisitet:	65 %
Gruppe 1:	8	11	19	Sensitivitet:	58 %
Totalt:	39	28	67	CCReq:	61 %
Nr. 14	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	58 %
Gruppe 0:	37	11	48	Spesifisitet:	77 %
Gruppe 1:	17	2	19	Sensitivitet:	11 %
Totalt:	54	13	67	CCReq:	44 %
Nr. 15	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	66 %
Gruppe 0:	43	5	48	Spesifisitet:	90 %
Gruppe 1:	18	1	19	Sensitivitet:	5 %
Totalt:	61	6	67	CCReq:	47 %
Nr. 16	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	67 %
Gruppe 0:	39	9	48	Spesifisitet:	81 %
Gruppe 1:	13	6	19	Sensitivitet:	32 %
Totalt:	52	15	67	CCReq:	56 %
Nr. 17	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	45 %
Gruppe 0:	14	34	48	Spesifisitet:	29 %
Gruppe 1:	3	16	19	Sensitivitet:	84 %
Totalt:	17	50	67	CCReq:	57 %
Nr. 18	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	49 %
Gruppe 0:	22	26	48	Spesifisitet:	46 %
Gruppe 1:	8	11	19	Sensitivitet:	58 %
Totalt:	30	37	67	CCReq:	52 %

Tabell 7.19: Resultater for hver VAR_c -gruppe, etter bruk av $VAR_{c16,2}$ på treningssettet i L23.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	34 %
Gruppe 0:	7	42	49	Spesifisitet:	14 %
Gruppe 1:	2	16	18	Sensitivitet:	89 %
Totalt:	9	58	67	CCReq:	52 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	45 %
Gruppe 0:	14	35	49	Spesifisitet:	29 %
Gruppe 1:	2	16	18	Sensitivitet:	89 %
Totalt:	16	51	67	CCReq:	59 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	37 %
Gruppe 0:	10	39	49	Spesifisitet:	20 %
Gruppe 1:	3	15	18	Sensitivitet:	83 %
Totalt:	13	54	67	CCReq:	52 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	40 %
Gruppe 0:	12	37	49	Spesifisitet:	24 %
Gruppe 1:	3	15	18	Sensitivitet:	83 %
Totalt:	15	52	67	CCReq:	54 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	40 %
Gruppe 0:	11	38	49	Spesifisitet:	22 %
Gruppe 1:	2	16	18	Sensitivitet:	89 %
Totalt:	13	54	67	CCReq:	56 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	39 %
Gruppe 0:	10	39	49	Spesifisitet:	20 %
Gruppe 1:	2	16	18	Sensitivitet:	89 %
Totalt:	12	55	67	CCReq:	55 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	40 %
Gruppe 0:	10	39	49	Spesifisitet:	20 %
Gruppe 1:	1	17	18	Sensitivitet:	94 %
Totalt:	11	56	67	CCReq:	57 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	36 %
Gruppe 0:	8	41	49	Spesifisitet:	16 %
Gruppe 1:	2	16	18	Sensitivitet:	89 %
Totalt:	10	57	67	CCReq:	53 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	40 %
Gruppe 0:	12	37	49	Spesifisitet:	24 %
Gruppe 1:	3	15	18	Sensitivitet:	83 %
Totalt:	15	52	67	CCReq:	54 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	61 %
Gruppe 0:	32	17	49	Spesifisitet:	65 %
Gruppe 1:	9	9	18	Sensitivitet:	50 %
Totalt:	41	26	67	CCReq:	58 %

Tabell 7.20: Resultater for hver VAR_c -gruppe, etter bruk av $VAR_{c16,2}$ på testsettet i L23

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	72 %
Gruppe 0:	48	0	48	Spesifisitet:	100 %
Gruppe 1:	19	0	19	Sensitivitet:	0 %
Totalt:	67	0	67	CCReq:	50 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	66 %
Gruppe 0:	44	4	48	Spesifisitet:	92 %
Gruppe 1:	19	0	19	Sensitivitet:	0 %
Totalt:	63	4	67	CCReq:	46 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	72 %
Gruppe 0:	47	1	48	Spesifisitet:	98 %
Gruppe 1:	18	1	19	Sensitivitet:	5 %
Totalt:	65	2	67	CCReq:	52 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	46 %
Gruppe 0:	18	30	48	Spesifisitet:	38 %
Gruppe 1:	6	13	19	Sensitivitet:	68 %
Totalt:	24	43	67	CCReq:	53 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	33 %
Gruppe 0:	4	44	48	Spesifisitet:	8 %
Gruppe 1:	1	18	19	Sensitivitet:	95 %
Totalt:	5	62	67	CCReq:	52 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	66 %
Gruppe 0:	43	5	48	Spesifisitet:	90 %
Gruppe 1:	18	1	19	Sensitivitet:	5 %
Totalt:	61	6	67	CCReq:	47 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	72 %
Gruppe 0:	48	0	48	Spesifisitet:	100 %
Gruppe 1:	19	0	19	Sensitivitet:	0 %
Totalt:	67	0	67	CCReq:	50 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	25 %
Gruppe 0:	17	31	48	Spesifisitet:	35 %
Gruppe 1:	19	0	19	Sensitivitet:	0 %
Totalt:	36	31	67	CCReq:	18 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	0 %
Gruppe 0:	0	48	48	Spesifisitet:	0 %
Gruppe 1:	19	0	19	Sensitivitet:	0 %
Totalt:	19	48	67	CCReq:	0 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	13 %
Gruppe 0:	8	40	48	Spesifisitet:	17 %
Gruppe 1:	18	1	19	Sensitivitet:	5 %
Totalt:	26	41	67	CCReq:	11 %

Tabell 7.21: Denne tabellen viser treningsresultatet for $LBP_{8,1}$ benyttet på de nedskalerte bildene i L23. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	42	7	49	Spesifisitet:	86 %
Gruppe 1:	13	5	18	Sensitivitet:	28 %
Totalt:	55	12	67	CCR _{eq} :	57 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	66 %
Gruppe 0:	29	20	49	Spesifisitet:	59 %
Gruppe 1:	3	15	18	Sensitivitet:	83 %
Totalt:	32	35	67	CCR _{eq} :	71 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	72 %
Gruppe 0:	39	10	49	Spesifisitet:	80 %
Gruppe 1:	9	9	18	Sensitivitet:	50 %
Totalt:	48	19	67	CCR _{eq} :	65 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	67 %
Gruppe 0:	42	7	49	Spesifisitet:	86 %
Gruppe 1:	15	3	18	Sensitivitet:	17 %
Totalt:	57	10	67	CCR _{eq} :	51 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	75 %
Gruppe 0:	41	8	49	Spesifisitet:	84 %
Gruppe 1:	9	9	18	Sensitivitet:	50 %
Totalt:	50	17	67	CCR _{eq} :	67 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	75 %
Gruppe 0:	37	12	49	Spesifisitet:	76 %
Gruppe 1:	5	13	18	Sensitivitet:	72 %
Totalt:	42	25	67	CCR _{eq} :	74 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	29	20	49	Spesifisitet:	59 %
Gruppe 1:	1	17	18	Sensitivitet:	94 %
Totalt:	30	37	67	CCR _{eq} :	77 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	63 %
Gruppe 0:	25	24	49	Spesifisitet:	51 %
Gruppe 1:	1	17	18	Sensitivitet:	94 %
Totalt:	26	41	67	CCR _{eq} :	73 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	48 %
Gruppe 0:	16	33	49	Spesifisitet:	33 %
Gruppe 1:	2	16	18	Sensitivitet:	89 %
Totalt:	18	49	67	CCR _{eq} :	61 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	72 %
Gruppe 0:	44	5	49	Spesifisitet:	90 %
Gruppe 1:	14	4	18	Sensitivitet:	22 %
Totalt:	58	9	67	CCR _{eq} :	56 %
Nr. 11	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	61 %
Gruppe 0:	35	14	49	Spesifisitet:	71 %
Gruppe 1:	12	6	18	Sensitivitet:	33 %
Totalt:	47	20	67	CCR _{eq} :	52 %
Nr. 12	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	41	8	49	Spesifisitet:	84 %
Gruppe 1:	13	5	18	Sensitivitet:	28 %
Totalt:	54	13	67	CCR _{eq} :	56 %
Nr. 13	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	73 %
Gruppe 0:	39	10	49	Spesifisitet:	80 %
Gruppe 1:	8	10	18	Sensitivitet:	56 %
Totalt:	47	20	67	CCR _{eq} :	68 %
Nr. 14	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	49 %
Gruppe 0:	16	33	49	Spesifisitet:	33 %
Gruppe 1:	1	17	18	Sensitivitet:	94 %
Totalt:	17	50	67	CCR _{eq} :	64 %
Nr. 15	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	34	15	49	Spesifisitet:	69 %
Gruppe 1:	6	12	18	Sensitivitet:	67 %
Totalt:	40	27	67	CCR _{eq} :	68 %
Nr. 16	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	43	6	49	Spesifisitet:	88 %
Gruppe 1:	14	4	18	Sensitivitet:	22 %
Totalt:	57	10	67	CCR _{eq} :	55 %
Nr. 17	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	0 %
Gruppe 0:	0	49	49	Spesifisitet:	0 %
Gruppe 1:	18	0	18	Sensitivitet:	0 %
Totalt:	18	49	67	CCR _{eq} :	0 %

Tabell 7.22: Denne tabellen viser testresultatet for $LBP_{8,1}$ benyttet på de nedskalerte bildene i L23. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCRReq.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	63 %
Gruppe 0:	36	12	48	Spesifisitet:	75 %
Gruppe 1:	13	6	19	Sensitivitet:	32 %
Totalt:	49	18	67	CCRReq:	53 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	29	19	48	Spesifisitet:	60 %
Gruppe 1:	2	17	19	Sensitivitet:	89 %
Totalt:	31	36	67	CCRReq:	75 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	58 %
Gruppe 0:	32	16	48	Spesifisitet:	67 %
Gruppe 1:	12	7	19	Sensitivitet:	37 %
Totalt:	44	23	67	CCRReq:	52 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	61 %
Gruppe 0:	39	9	48	Spesifisitet:	81 %
Gruppe 1:	17	2	19	Sensitivitet:	11 %
Totalt:	56	11	67	CCRReq:	46 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	64 %
Gruppe 0:	36	12	48	Spesifisitet:	75 %
Gruppe 1:	12	7	19	Sensitivitet:	37 %
Totalt:	48	19	67	CCRReq:	56 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	75 %
Gruppe 0:	36	12	48	Spesifisitet:	75 %
Gruppe 1:	5	14	19	Sensitivitet:	74 %
Totalt:	41	26	67	CCRReq:	74 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	66 %
Gruppe 0:	26	22	48	Spesifisitet:	54 %
Gruppe 1:	1	18	19	Sensitivitet:	95 %
Totalt:	27	40	67	CCRReq:	74 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	66 %
Gruppe 0:	26	22	48	Spesifisitet:	54 %
Gruppe 1:	1	18	19	Sensitivitet:	95 %
Totalt:	27	40	67	CCRReq:	74 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	19	29	48	Spesifisitet:	40 %
Gruppe 1:	1	18	19	Sensitivitet:	95 %
Totalt:	20	47	67	CCRReq:	67 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	63 %
Gruppe 0:	39	9	48	Spesifisitet:	81 %
Gruppe 1:	16	3	19	Sensitivitet:	16 %
Totalt:	55	12	67	CCRReq:	49 %
Nr. 11	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	51 %
Gruppe 0:	26	22	48	Spesifisitet:	54 %
Gruppe 1:	11	8	19	Sensitivitet:	42 %
Totalt:	37	30	67	CCRReq:	48 %
Nr. 12	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	57 %
Gruppe 0:	32	16	48	Spesifisitet:	67 %
Gruppe 1:	13	6	19	Sensitivitet:	32 %
Totalt:	45	22	67	CCRReq:	49 %
Nr. 13	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	66 %
Gruppe 0:	36	12	48	Spesifisitet:	75 %
Gruppe 1:	11	8	19	Sensitivitet:	42 %
Totalt:	47	20	67	CCRReq:	59 %
Nr. 14	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	40 %
Gruppe 0:	10	38	48	Spesifisitet:	21 %
Gruppe 1:	2	17	19	Sensitivitet:	89 %
Totalt:	12	55	67	CCRReq:	55 %
Nr. 15	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	42 %
Gruppe 0:	24	24	48	Spesifisitet:	50 %
Gruppe 1:	15	4	19	Sensitivitet:	21 %
Totalt:	39	28	67	CCRReq:	36 %
Nr. 16	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	42	6	48	Spesifisitet:	88 %
Gruppe 1:	15	4	19	Sensitivitet:	21 %
Totalt:	57	10	67	CCRReq:	54 %
Nr. 17	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	0 %
Gruppe 0:	0	48	48	Spesifisitet:	0 %
Gruppe 1:	19	0	19	Sensitivitet:	0 %
Totalt:	19	48	67	CCRReq:	0 %

Tabell 7.23: Denne tabellen viser treningsresultatet for VAR_c benyttet på de nedskalerte cellekjerne-bildene i L23. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCR_{eq}.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	33	16	49	Spesifisitet:	67 %
Gruppe 1:	4	14	18	Sensitivitet:	78 %
Totalt:	37	30	67	CCR _{eq} :	73 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	39 %
Gruppe 0:	11	38	49	Spesifisitet:	22 %
Gruppe 1:	3	15	18	Sensitivitet:	83 %
Totalt:	14	53	67	CCR _{eq} :	53 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	61 %
Gruppe 0:	29	20	49	Spesifisitet:	59 %
Gruppe 1:	6	12	18	Sensitivitet:	67 %
Totalt:	35	32	67	CCR _{eq} :	63 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	67 %
Gruppe 0:	37	12	49	Spesifisitet:	76 %
Gruppe 1:	10	8	18	Sensitivitet:	44 %
Totalt:	47	20	67	CCR _{eq} :	60 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	77.6119 %
Gruppe 0:	41	8	49	Spesifisitet:	84 %
Gruppe 1:	7	11	18	Sensitivitet:	61 %
Totalt:	48	19	67	CCR _{eq} :	72 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	79 %
Gruppe 0:	42	7	49	Spesifisitet:	86 %
Gruppe 1:	7	11	18	Sensitivitet:	61 %
Totalt:	49	18	67	CCR _{eq} :	73 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	81 %
Gruppe 0:	43	6	49	Spesifisitet:	88 %
Gruppe 1:	7	11	18	Sensitivitet:	61 %
Totalt:	50	17	67	CCR _{eq} :	74 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	36	13	49	Spesifisitet:	73 %
Gruppe 1:	8	10	18	Sensitivitet:	56 %
Totalt:	44	23	67	CCR _{eq} :	65 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	31	18	49	Spesifisitet:	63 %
Gruppe 1:	9	9	18	Sensitivitet:	50 %
Totalt:	40	27	67	CCR _{eq} :	57 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	60 %
Gruppe 0:	30	19	49	Spesifisitet:	61 %
Gruppe 1:	8	10	18	Sensitivitet:	56 %
Totalt:	38	29	67	CCR _{eq} :	58 %

Tabell 7.24: Denne tabellen viser testresultatet for VAR_c benyttet på de nedskalerte cellekjerne-bildene i L23. Resultatene består av forvirringsmatriser, mål på sensitivitet og spesifisitet samt CCRReq.

Nr. 1	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	75 %
Gruppe 0:	33	15	48	Spesifisitet:	69 %
Gruppe 1:	2	17	19	Sensitivitet:	89 %
Totalt:	35	32	67	CCRReq:	79 %
Nr. 2	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	31 %
Gruppe 0:	7	41	48	Spesifisitet:	15 %
Gruppe 1:	5	14	19	Sensitivitet:	74 %
Totalt:	12	55	67	CCRReq:	44 %
Nr. 3	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	55 %
Gruppe 0:	27	21	48	Spesifisitet:	56 %
Gruppe 1:	9	10	19	Sensitivitet:	53 %
Totalt:	36	31	67	CCRReq:	54 %
Nr. 4	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	67 %
Gruppe 0:	39	9	48	Spesifisitet:	81 %
Gruppe 1:	13	6	19	Sensitivitet:	32 %
Totalt:	52	15	67	CCRReq:	56 %
Nr. 5	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	63 %
Gruppe 0:	33	15	48	Spesifisitet:	69 %
Gruppe 1:	10	9	19	Sensitivitet:	47 %
Totalt:	43	24	67	CCRReq:	58 %
Nr. 6	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	67 %
Gruppe 0:	34	14	48	Spesifisitet:	71 %
Gruppe 1:	8	11	19	Sensitivitet:	58 %
Totalt:	42	25	67	CCRReq:	64 %
Nr. 7	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	69 %
Gruppe 0:	37	11	48	Spesifisitet:	77 %
Gruppe 1:	10	9	19	Sensitivitet:	47 %
Totalt:	47	20	67	CCRReq:	62 %
Nr. 8	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	70 %
Gruppe 0:	37	11	48	Spesifisitet:	77 %
Gruppe 1:	9	10	19	Sensitivitet:	53 %
Totalt:	46	21	67	CCRReq:	65 %
Nr. 9	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	64 %
Gruppe 0:	34	14	48	Spesifisitet:	71 %
Gruppe 1:	10	9	19	Sensitivitet:	47 %
Totalt:	44	23	67	CCRReq:	59 %
Nr. 10	Gruppe 0	Gruppe 1	Totalt	Total andel riktig klassifisering:	61 %
Gruppe 0:	33	15	48	Spesifisitet:	69 %
Gruppe 1:	11	8	19	Sensitivitet:	42 %
Totalt:	44	23	67	CCRReq:	55 %

Bibliografi

- [1] A. Rényi. On measures of information and entropy. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability 1960*, pages 547–561, 1961.
- [2] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2037–2041, 2006.
- [3] F. Albrechtsen. Privat kommunikasjon.
- [4] F. Albrechtsen. Rotation and Mirror Invariant LBP Texture Patterns. unpublished note, January 2012.
- [5] F. Albrechtsen and B. Nielsen. Texture Classification based on Cooccurrence of Gray Level Run Length Matrices. *Australian Journal of Intelligent Information Processing Systems*, 6:38–45, 2000.
- [6] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 4 1943.
- [7] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, 1966.
- [8] S. Cha. Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1:300–307, 2007.
- [9] A. Chu, C.M. Sehgal, and J.F. Greenleaf. Use of gray value distribution of run lengths for texture analysis. *Pattern Recognition Letters*, 11(6):415–419, 1990.
- [10] R. W. Connors, M. M. Trivedi, and C. A. Harlow. Segmentation of a High-Resolution Urban Scene Using Texture Operators. *Computer Vision, Graphics and Image Processing*, 1984.
- [11] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [12] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

- [13] T.M. Cover. The best two independent measurements are not the two best. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-4(1):116–117, 1974.
- [14] D. W. Hedley. DNA Analysis from Paraffin-Embedded Blocks. *Methods in Cell Biology*, 41:231–240, 1994.
- [15] H. E. Danielsen. Privat kommunikasjon.
- [16] K. G. Derpanis. The Bhattacharyya Measure, 2008.
- [17] J. L. Devore and K. N. Berk. *Modern Mathematical Statistics with Applications*. Duxbury, 2007.
- [18] L. Du, X. You, H. Xu, Z. Gao, and Y. Tang. Wavelet domain local binary pattern features for writer identification. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3691–3694. IEEE, 2010.
- [19] S. F. Ershad. Texture Classification Approach Based on Combination of Edge & Co-occurrence and Local Binary Pattern. *arXiv preprint arXiv:1203.4855*, 2012.
- [20] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, second edition, 1990.
- [21] M. M. Galloway. Texture analysis using gray level run lengths. *Computer graphics and image processing*, 4(2):172–179, 1975.
- [22] D. F. Gleason. Histologic grading of prostate cancer: a perspective. *Human pathology*, 23(3):273–279, 1992.
- [23] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, third, international edition, 2008.
- [24] D. J. Hand. *Discrimination and classification*. John Wiley, 1981.
- [25] R. M. Haralick. Statistical and Structural Approaches to Texture. *Proceedings of the IEEE*, 67(5):786–804, May 1979.
- [26] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural Features for Image Classification. *IEEE Transaction on Systems, Man and Cybernetics*, 1973.
- [27] D. Harwood, T. Ojala, M. Pietikäinen, S. Kelman, and L. Davis. Texture classification by center-symmetric auto-correlation, using Kullback discrimination of distributions. *Pattern Recognition Letters*, 16:1–10, January 1995.
- [28] D. C. He and L. Wang. Texture Feature Extraction From Texture Spectrum. In *Geoscience and Remote Sensing Symposium, 1990. IGARSS '90. 'Remote Sensing Science for the Nineties', 10th Annual International*, pages 1987–1990, May 1990.

- [29] G. Hughes. On the mean accuracy of statistical pattern recognizers. *Information Theory, IEEE Transactions on*, 14(1):55–63, Jan.
- [30] A. Jain and D. Zongker. Feature selection: evaluation, application, and small sample performance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(2):153–158, 1997.
- [31] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37, jan 2000.
- [32] T. Jørgensen, K. Yogesan, K. J. Tveter, F. Skjorten, and H. E. Danielsen. Nuclear texture analysis: a new prognostic tool in metastatic prostate cancer. *Cytometry*, 24(3):277–283, 1996.
- [33] L. Kanal. Patterns in pattern recognition: 1968-1974. *Information Theory, IEEE Transactions on*, 20(6):697–722, 1974.
- [34] R. L. Kashyap and A. Khotanzad. A Model-Based Method for Rotation Invariant Texture Classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(4):472–481, July 1986.
- [35] T. Leung and J. Malik. Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.
- [36] Y. Ma. Number Local Binary Pattern: An Extended Local Binary Pattern. In *Proceedings of the 2011 International Conference on Wavelet Analysis and Pattern Recognition*, July 2011.
- [37] P. C. Mahalanobis. On the generalised distance in statistics. In *Proceedings of the National Institute of Sciences of India*, volume 2, pages 49–55, 1936.
- [38] T. Mairinger, G. Mikuz, and A. Gschwendtner. Nuclear chromatin texture analysis of nonmalignant tissue can detect adjacent prostatic adenocarcinoma. *The Prostate*, 41(1):12–19, 1999.
- [39] T. Mänenpää. *The Local Binary Pattern Approach to Texture Analysis - Extensions and Applications*. PhD thesis, University of Oulu, 2003.
- [40] S. Marsland. *Machine learning: an algorithmic perspective*. Chapman & Hall/CRC, 2009.
- [41] G. J. McLachlan. Mahalanobis distance. *Resonance*, 4(6):20–26, 6 1999.
- [42] M. A. Mohamed, P. A. Greif, J. Diamond, O. Sharaf, P. Maxwell, R. Montironi, R. A.M. Young, and P. W. Hamilton. Epigenetic events, remodelling enzymes and their relationship to chromatin organization in prostatic intraepithelial neoplasia and prostatic adenocarcinoma. *BJU international*, 99(4):908–915, 2007.

- [43] B. Nielsen. Privat kommunikasjon.
- [44] B. Nielsen, F. Albrechtsen, and H. E. Danielsen. Low dimensional adaptive texture feature vectors from class distance and class difference matrices. *IEEE Trans Med Imaging*, 23:73–84, 2004.
- [45] B. Nielsen, F. Albrechtsen, and H. E. Danielsen. Low dimensional adaptive texture feature vectors from class distance and class difference matrices. *Medical Imaging, IEEE Transactions on*, 23(1):73–84, 2004.
- [46] B. Nielsen, F. Albrechtsen, and H. E. Danielsen. Statistical Nuclear Texture Analysis in Cancer Research: A Review of Methods and Applications. *Critical Reviews TM in Oncogenesis*, 2008.
- [47] B. Nielsen, F. Albrechtsen, and H.E. Danielsen. Fractal analysis of monolayer cell nuclei from two different prognostic classes of early ovarian cancer. In *Fractals in Biology and Medicine*, pages 175–186. Springer, 2005.
- [48] B. Nielsen, F. Albrechtsen, W. Kildal, V. M. Abeler, and G. B. Kristensen. The prognostic value of adaptive nuclear texture features from patient gray level entropy matrices in early stage ovarian cancer. *Analytical Cellular Pathology*, 35(4), 2012.
- [49] B. Nielsen, F. Albrechtsen, W. Kildal, and H. E. Danielsen. Prognostic classification of early ovarian cancer based on very low dimensionality adaptive texture feature vectors from cell nuclei from monolayers and histological sections. *Analytical Cellular Pathology*, 23(2):75–88, 2001.
- [50] B. Nielsen and H. E. Danielsen. Prognostic value of adaptive textural features-the effect of standardizing nuclear first-order gray level statistics and mixing information from nuclei having different area. *Analytical Cellular Pathology*, 28(3):85–95, 2006.
- [51] University of Oslo. <http://www.uio.no/studier/emner/matnat/ifi/INF4300/h11/undervisningsmateriale/INF4300-2011-f02-texture.pdf>.
- [52] University of Oulu. Outex Texture Database. <http://http://www.outex.oulu.fi/>.
- [53] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 582–585, October 1994.
- [54] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

- [55] M. Pedersen. *Image quality metrics for the evaluation of printing workflows*. PhD thesis, University of Oslo, 2011.
- [56] M. Pietikäinen, T. Ojala, and Z. Xu. Rotation-invariant Texture Classification Using Feature Distributions. *Pattern Recognition*, 33(1):43 – 52, 2000.
- [57] M. Pietikäinen and G. Zhao. Tutorial - Local Texture Descriptors in Computer Vision, September 2009.
- [58] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern recognition letters*, 15(11):1119–1125, 1994.
- [59] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover’s Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40:99–121, 2000.
- [60] B. Sujatha, Dr V. VijayaKumar, and Dr P. Harini. A New Logical Compact LBP Co-Occurrence Matrix for Texture Analysis. *International Journal of Scientific & Engineering Research*, 3(2), 2012.
- [61] T. Ojala and M. Pietikäinen and D. Harwood. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, volume 1, pages 582–585, October 1994.
- [62] H.J. Tanke and E.M. Van Ingen. A reliable feulgen-acriflavine-so2 staining procedure for quantitative dna measurements. *Journal of Histochemistry & Cytochemistry*, 28(9):1007–1013, 1980.
- [63] M. Unser. Sum and Difference Histograms for Texture Classification. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1986.
- [64] R. F. Walker, P. T. Jackway, and I. D. Longstaff. Recent developments in the use of the co-occurrence matrix for texture recognition. In *Proceedings of the 13th International Conference on Digital Signal Processing*, volume 1, pages 63–65, 1997.
- [65] Wikipedia. http://en.wikipedia.org/wiki/Entropic_uncertainty.
- [66] Wikipedia. http://en.wikipedia.org/wiki/Correlation_and_dependence.
- [67] Wikipedia. http://en.wikipedia.org/wiki/Sensitivity_and_specificity.
- [68] K. Yogesan, T. Jørgensen, F. Albregtsen, K. J. Tveter, and H. E. Danielsen. Entropy-based texture analysis of chromatin structure in advanced prostate cancer. *Cytometry*, 24, 1996.

- [69] D. Zongker and A. Jain. Algorithms for feature selection: An evaluation.
In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 2, pages 18–22 vol.2, 1996.

